# Reproducible research with R

19th June 2013

# Outline

## Part I – version control

- Introduction
  - Reproducibility of analysis
  - Version control
- Project design
  - What is version controlled?
  - Data persistence
- Using version control with R
  - Setting up a project
  - Using Git with R

# Outline

## Part II – Report generation

- Introduction
  - Report generation
  - Tools
- Markdown – HTML output
  - Markdown language
  - Exercises with R
- Sweave and LateX – PDF output
  - Latex
  - Exercises with R

# Outline

## Part II – Report generation

# Outline

## Part II – Report generation

- Introduction
  - Report generation
  - Tools
- Markdown – HTML output
  - Markdown language
  - Exercises with R
- Sweave and LateX – PDF output
  - Latex
  - Exercises with R

# Outline

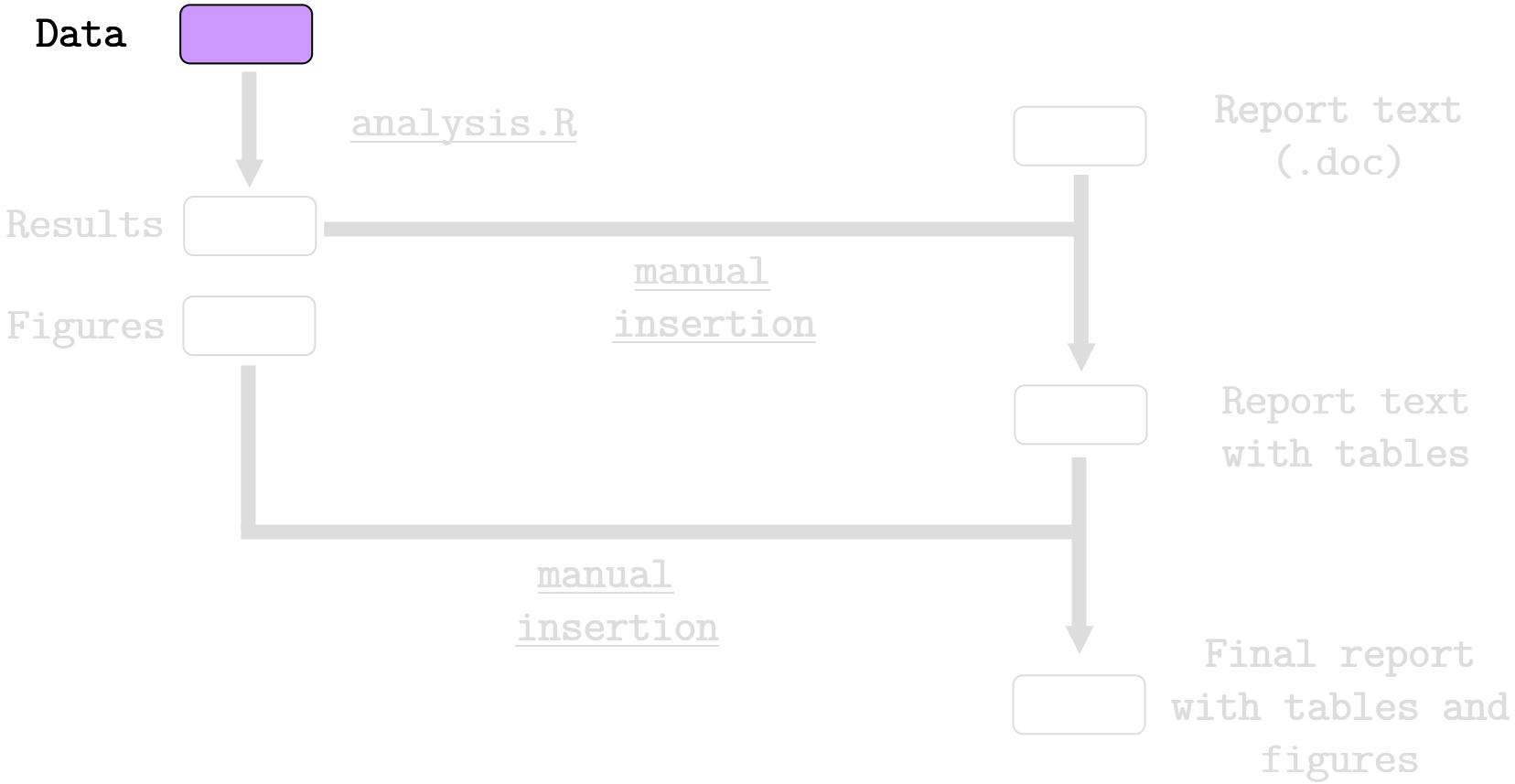
## Part II – Report generation

- Introduction
  - Report generation
  - Tools
- Markdown – HTML output
  - Markdown language
  - Exercises with R
- Sweave and LaTeX – PDF output
  - Latex
  - Exercises with R

# Part II
# Report generation

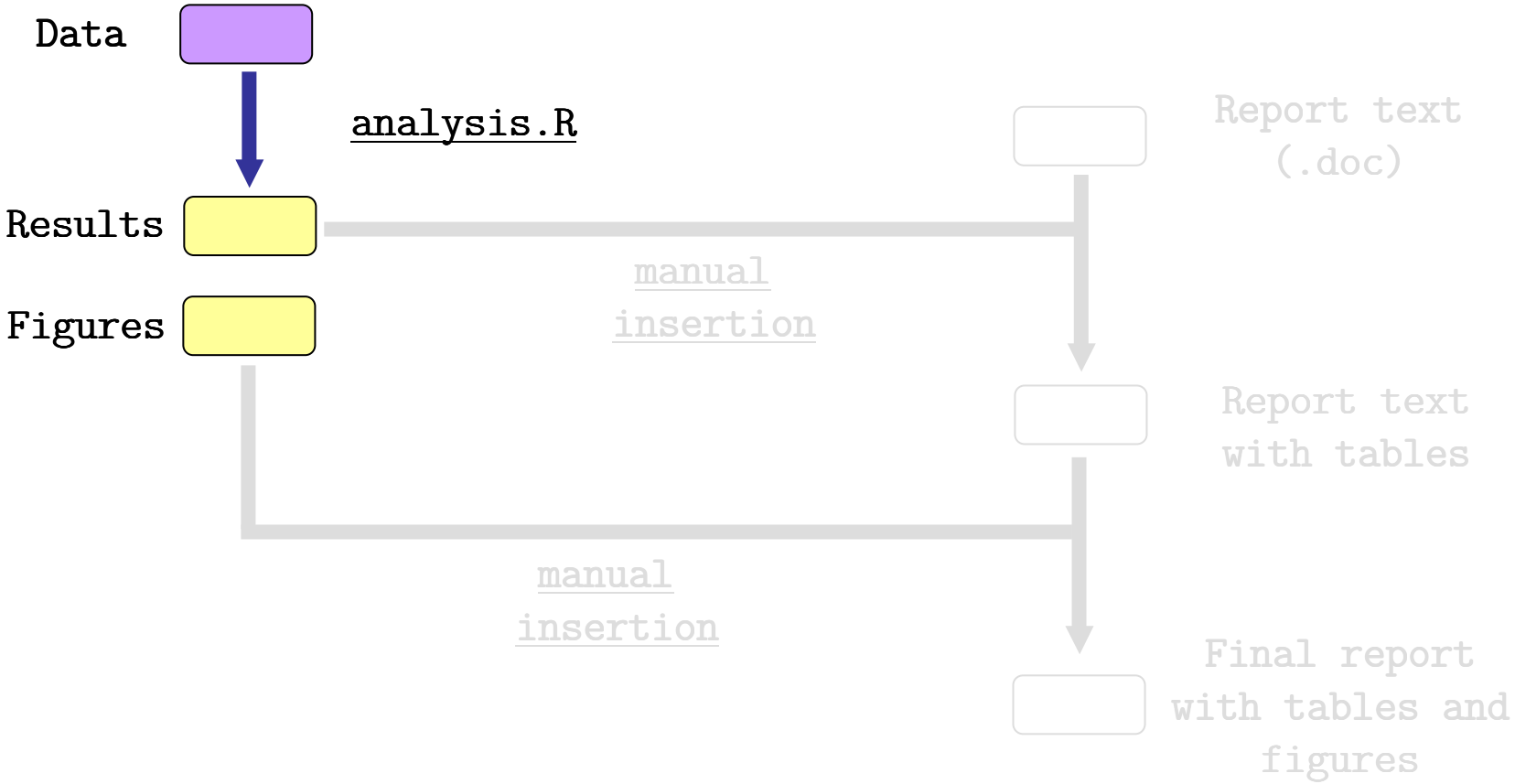
# The usual approach





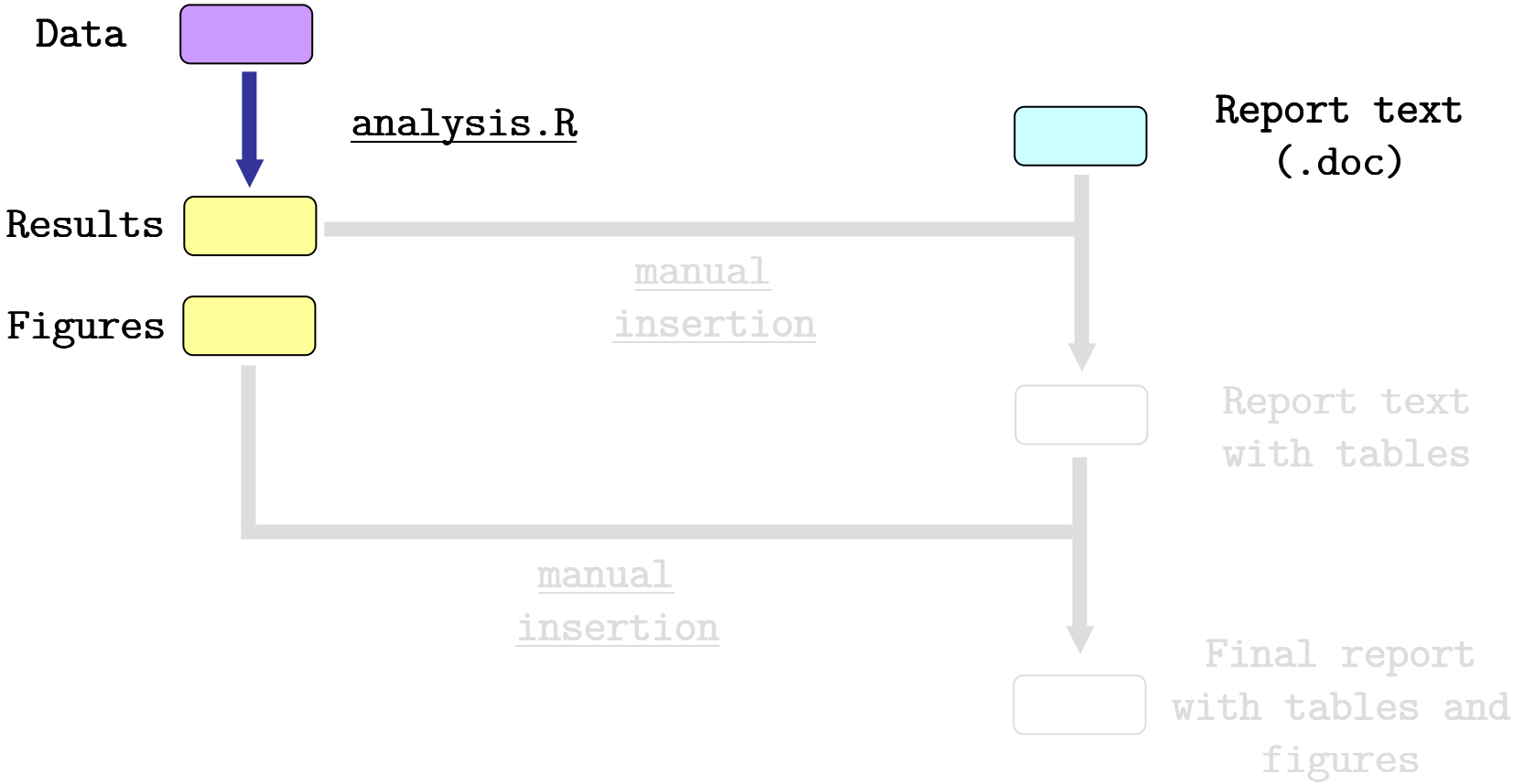
# The usual approach





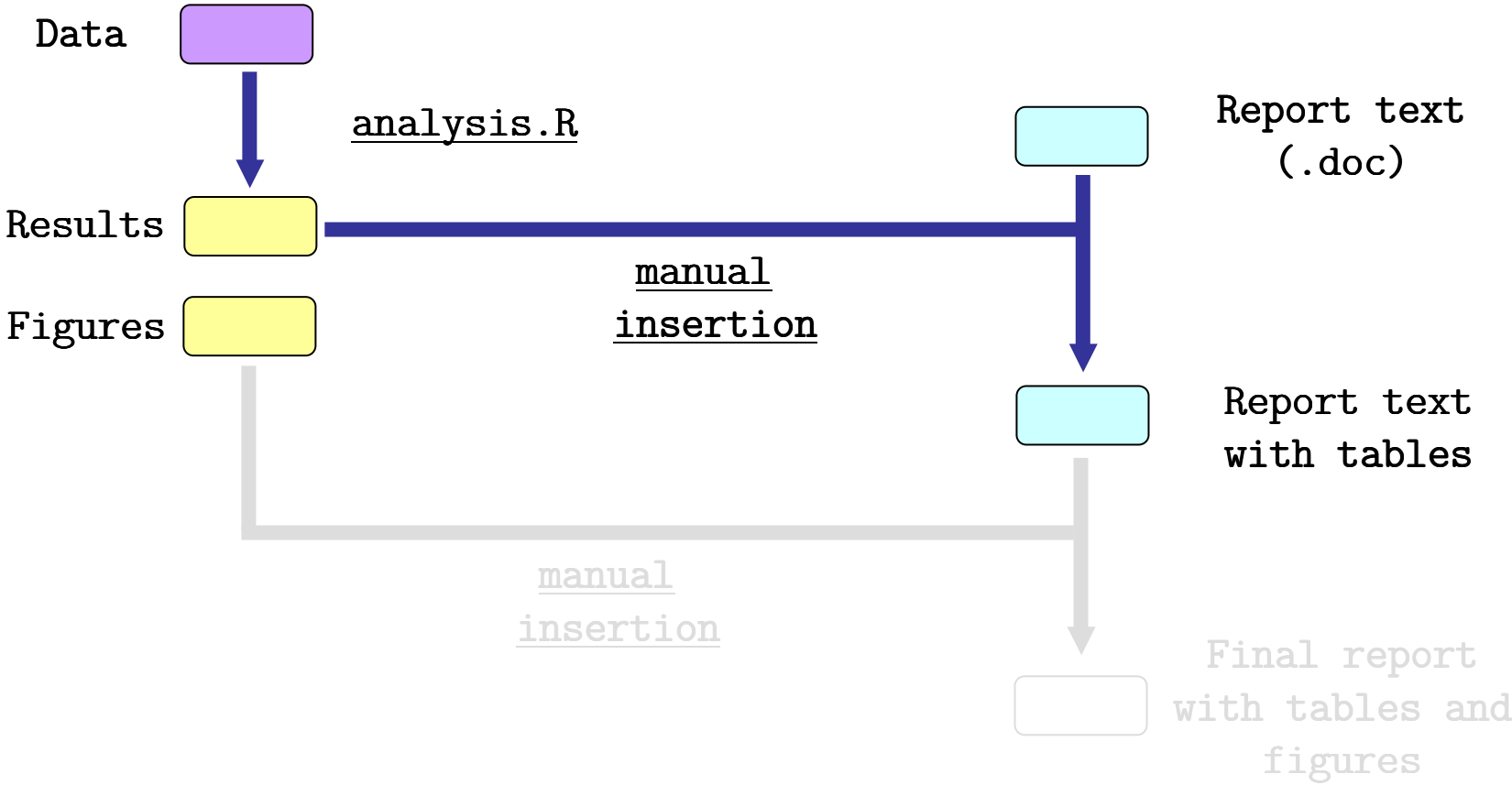
# The usual approach





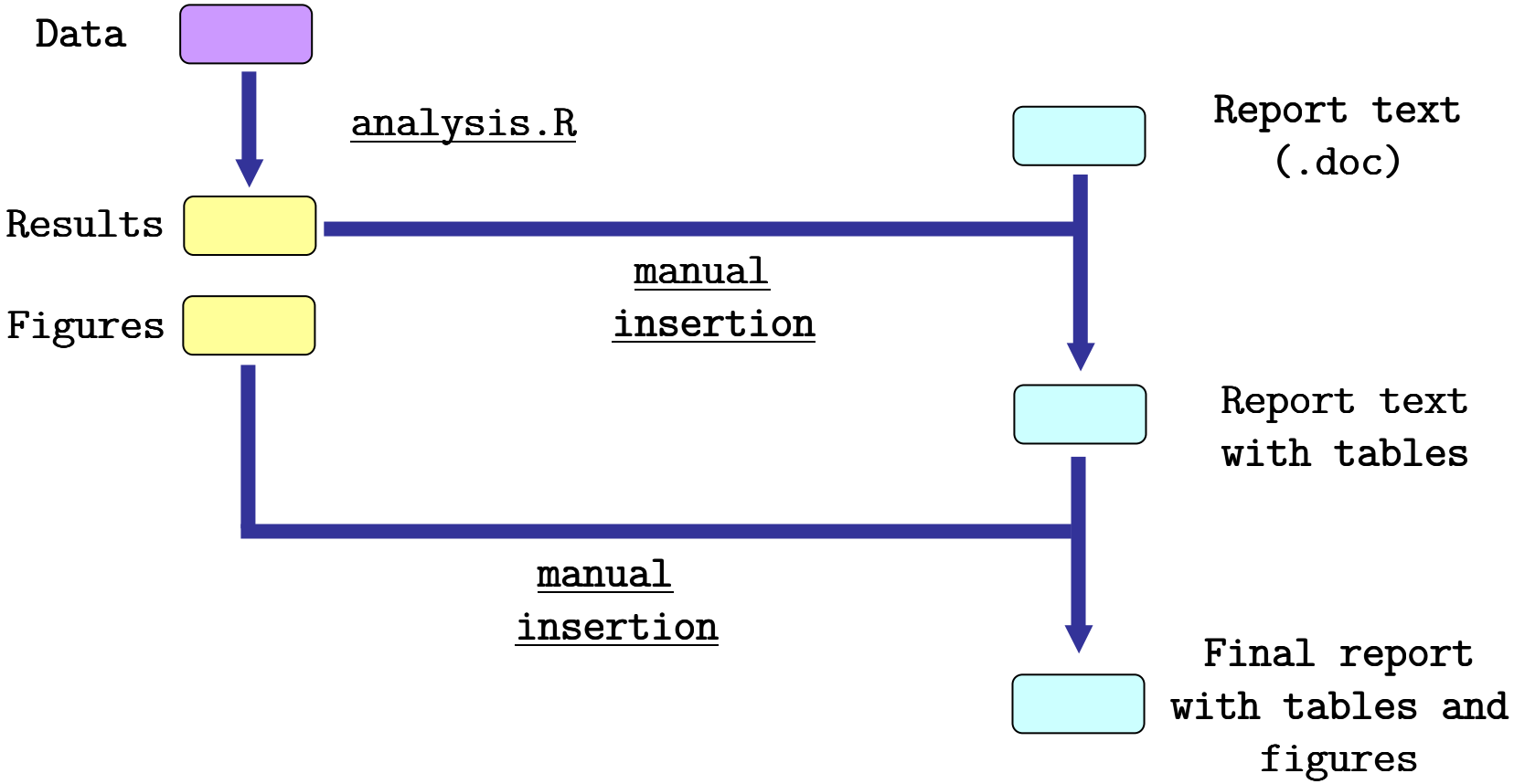
# The usual approach





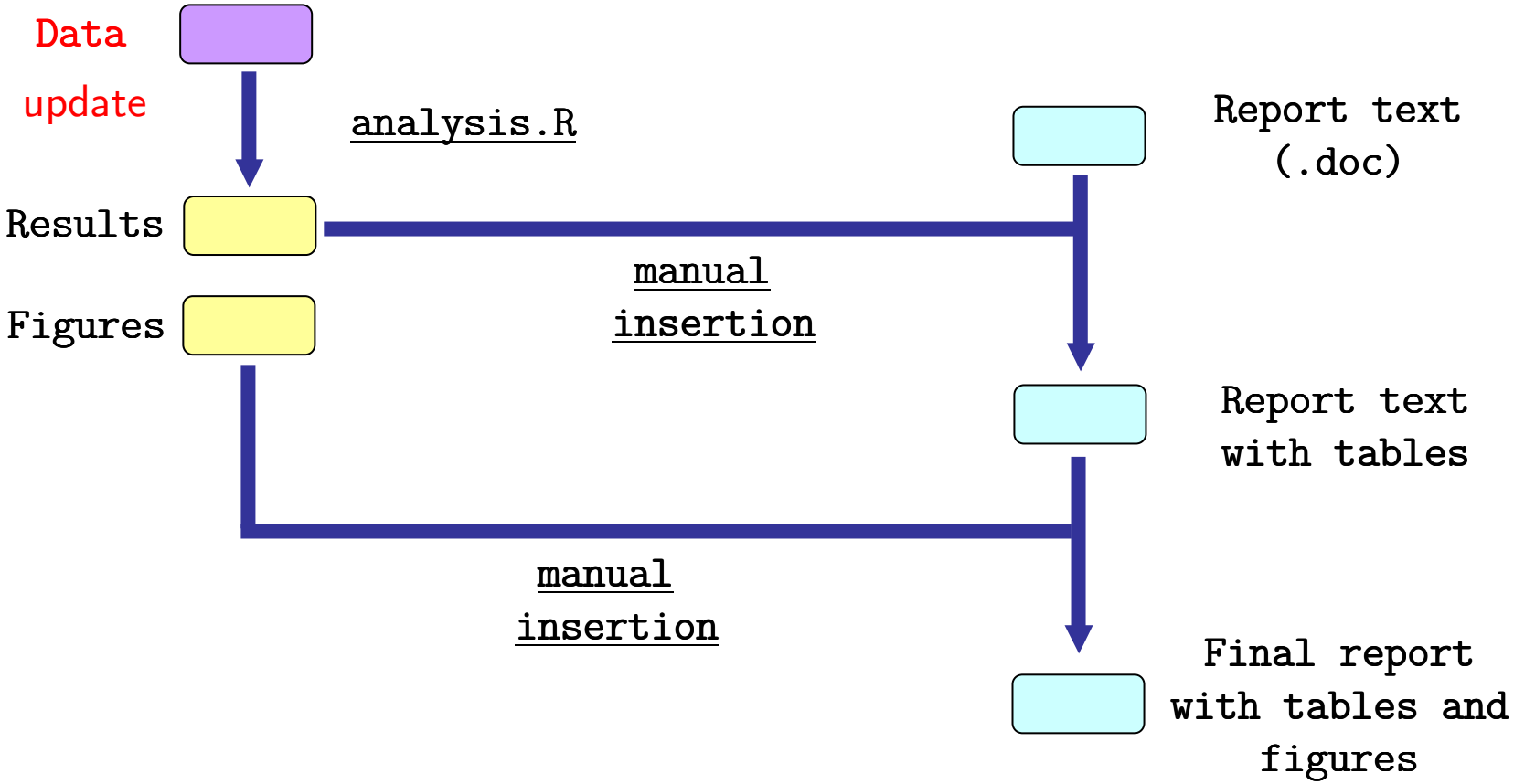
# The usual approach





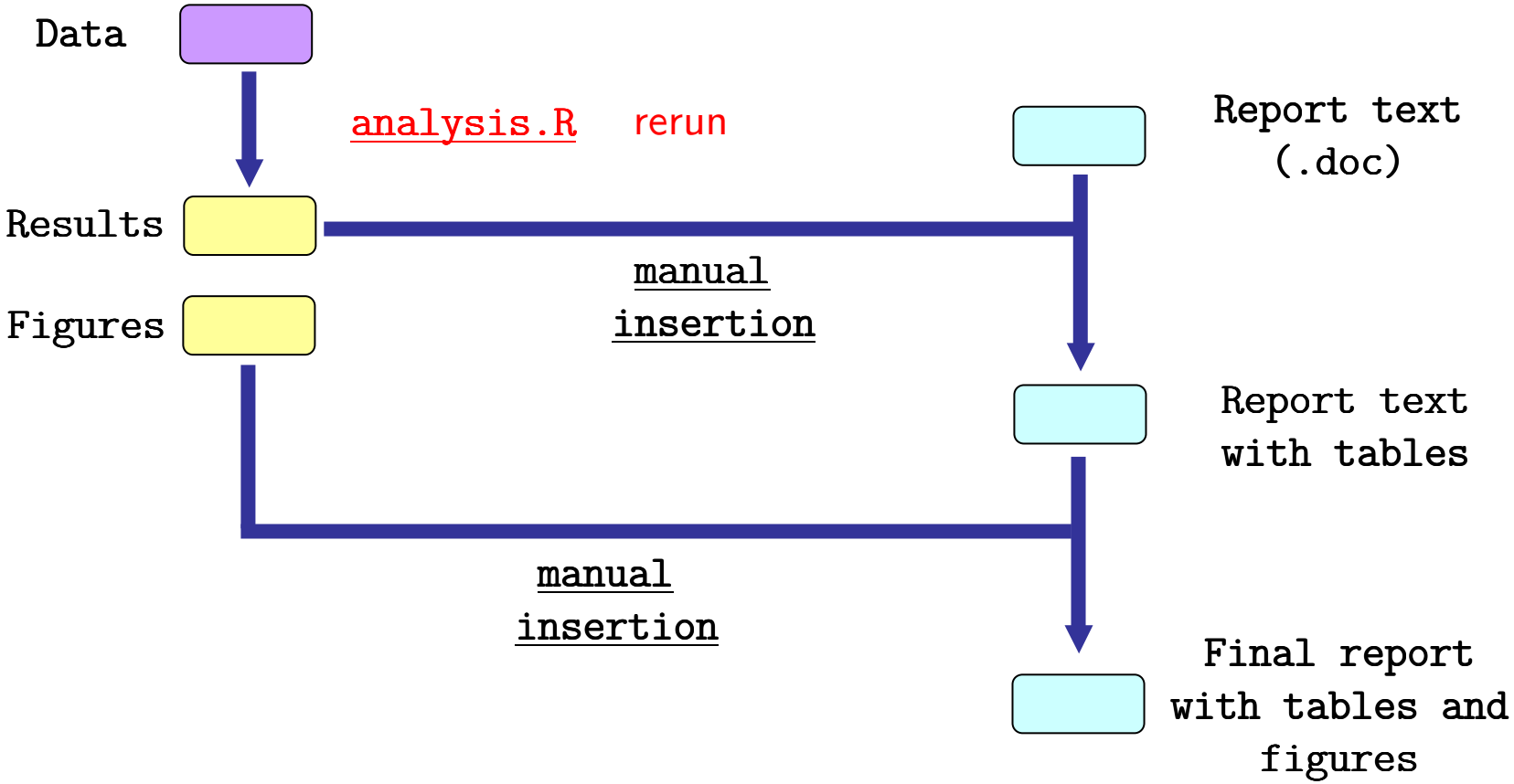
# The usual approach





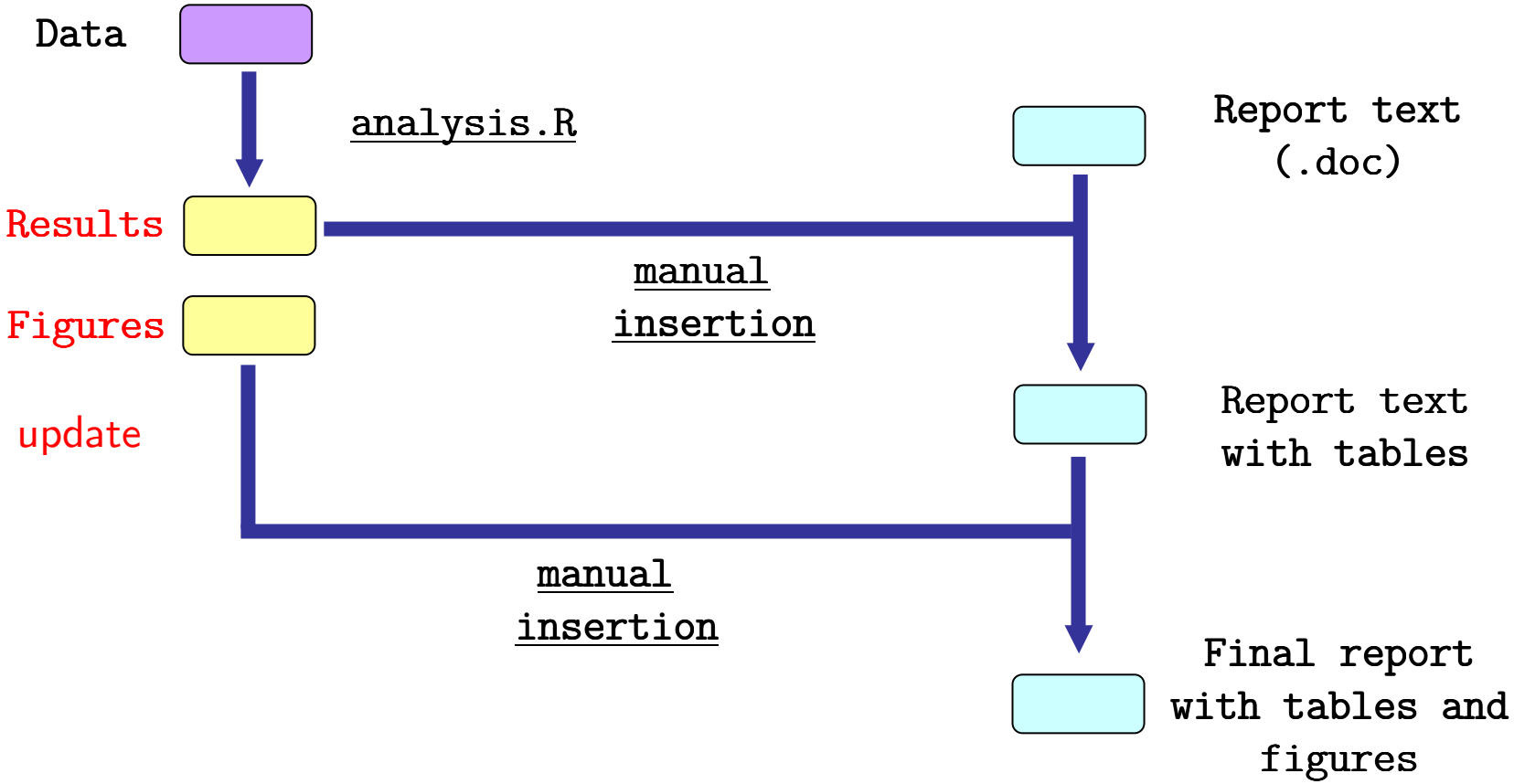
# The usual approach





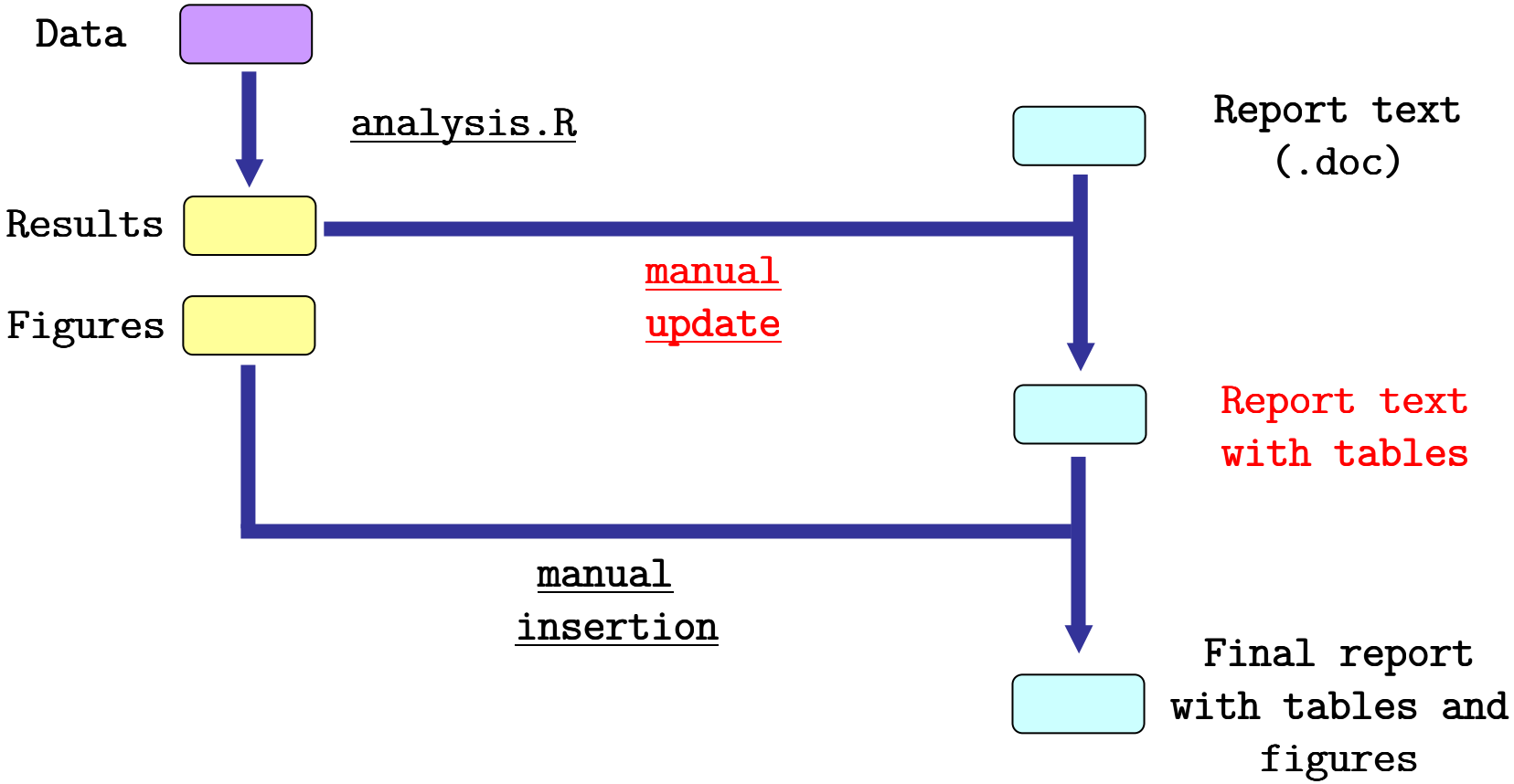
# The usual approach





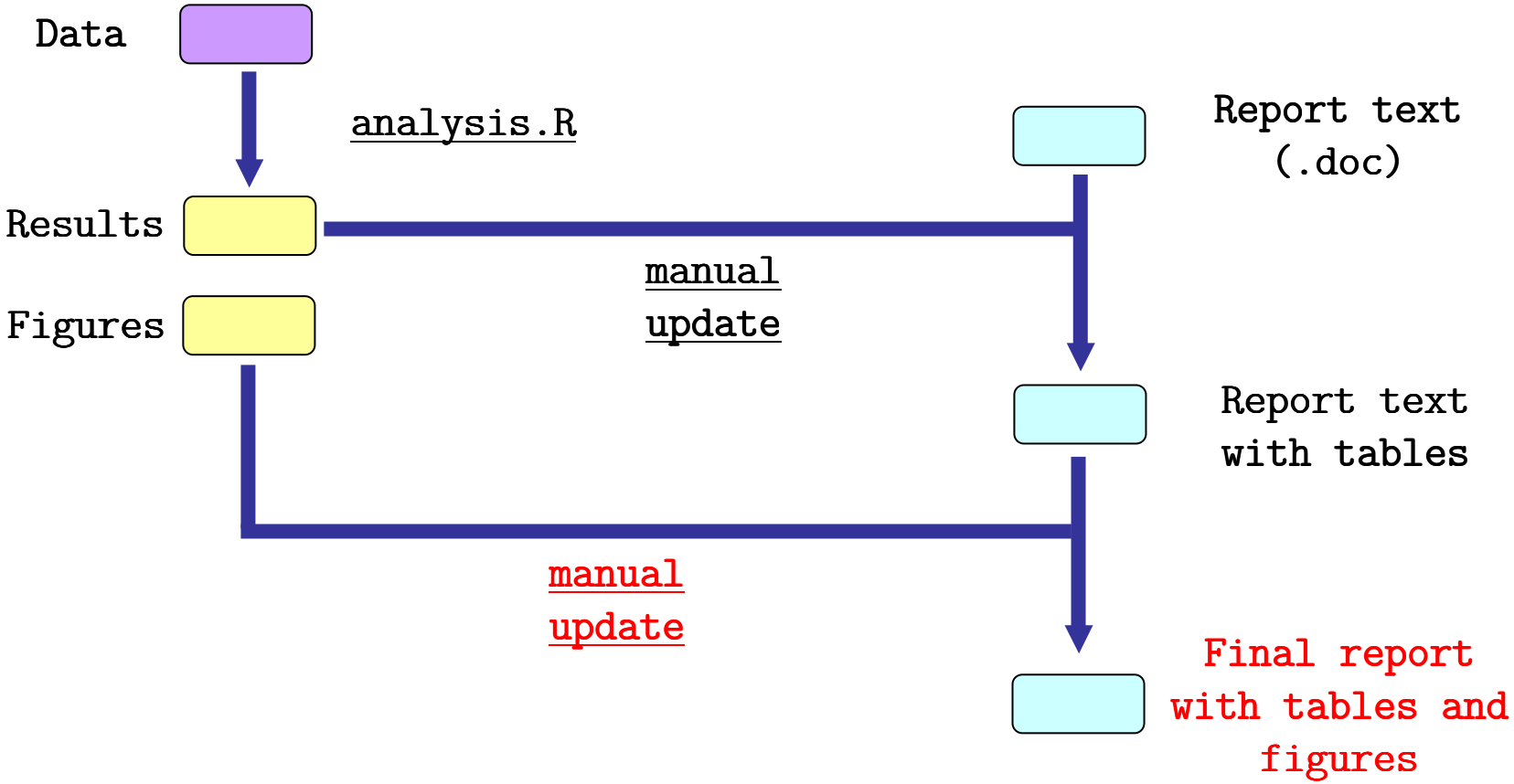
# The usual approach





# The usual approach





# The usual approach



- Manual update is time consuming.
- It is also error prone.
- The analysis and the report are separated in different files.

# The usual approach



- Manual update is time consuming.
- It is also error prone.
- The analysis and the report are separated in different files.

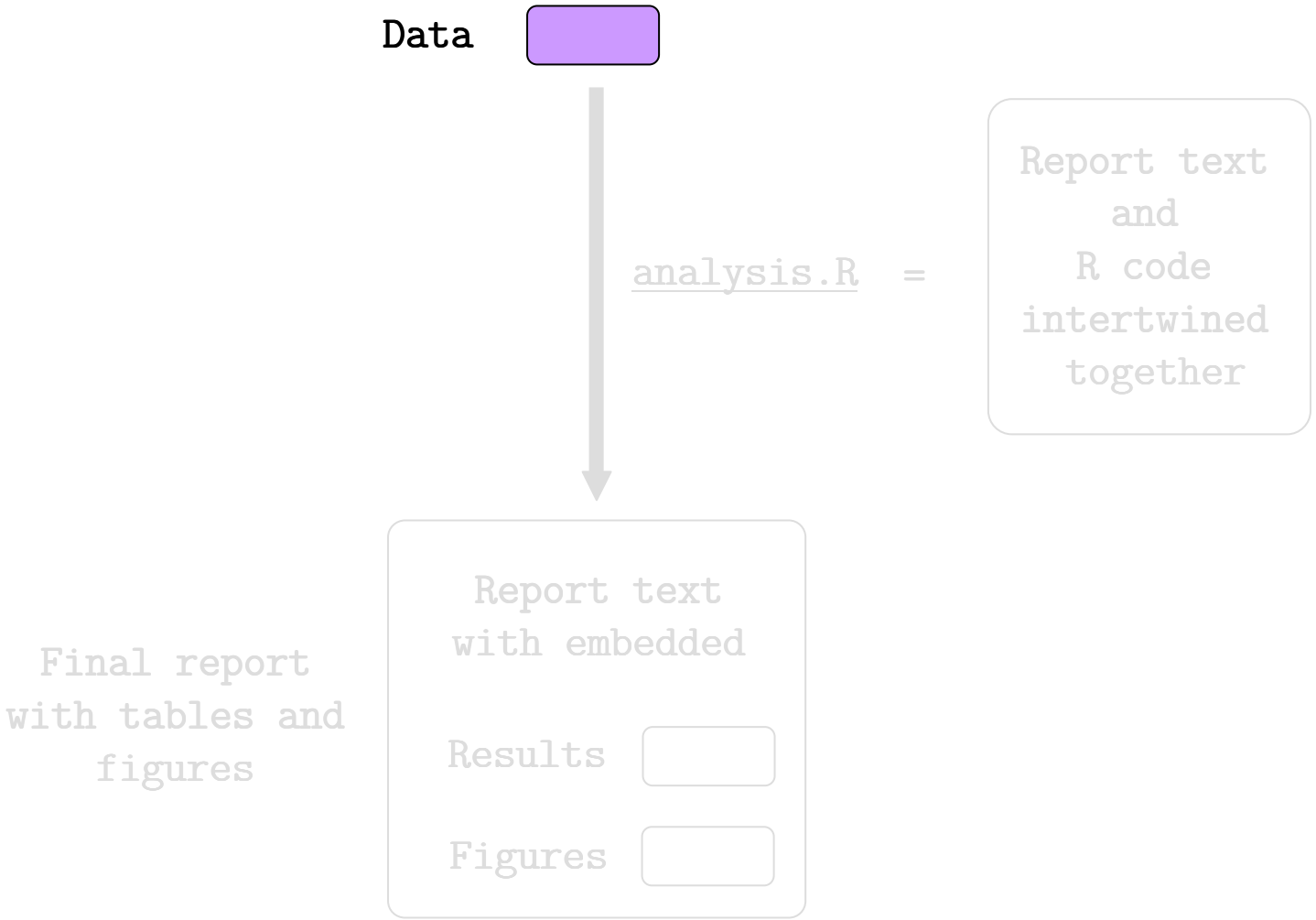
# The usual approach



- Manual update is time consuming.
- It is also error prone.
- The analysis and the report are separated in different files.

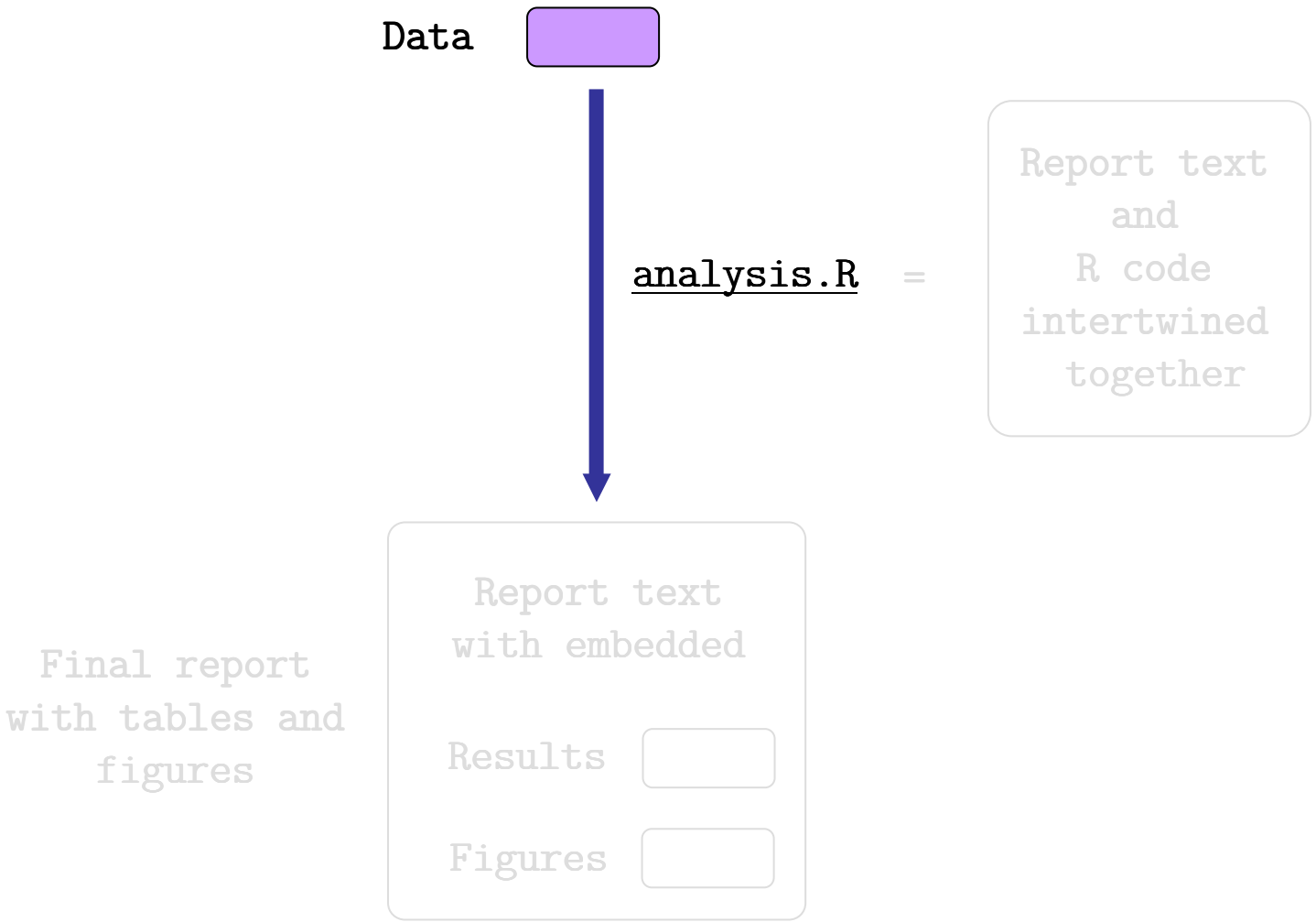
# Automatic report generation





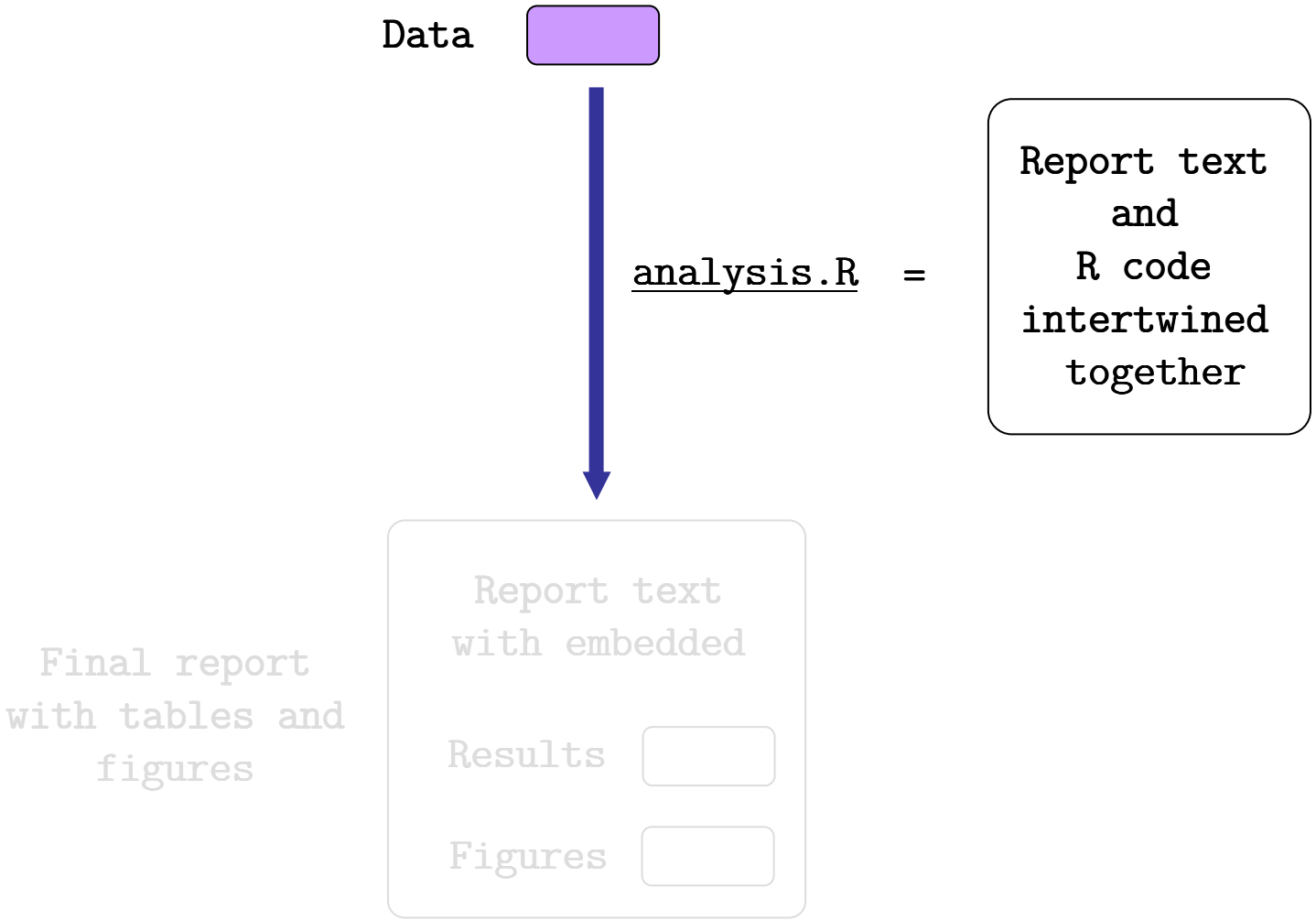
# Automatic report generation





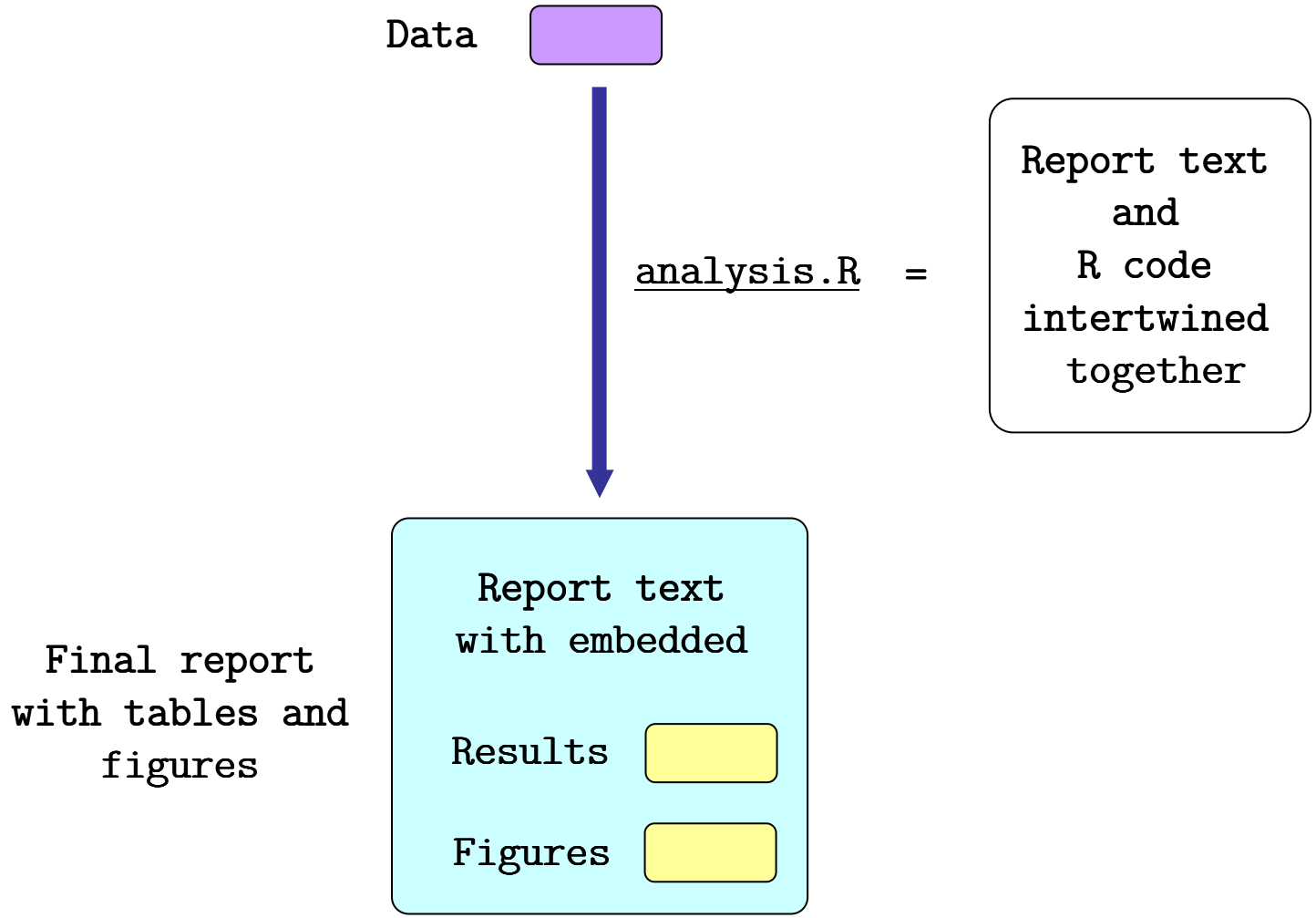
# Automatic report generation





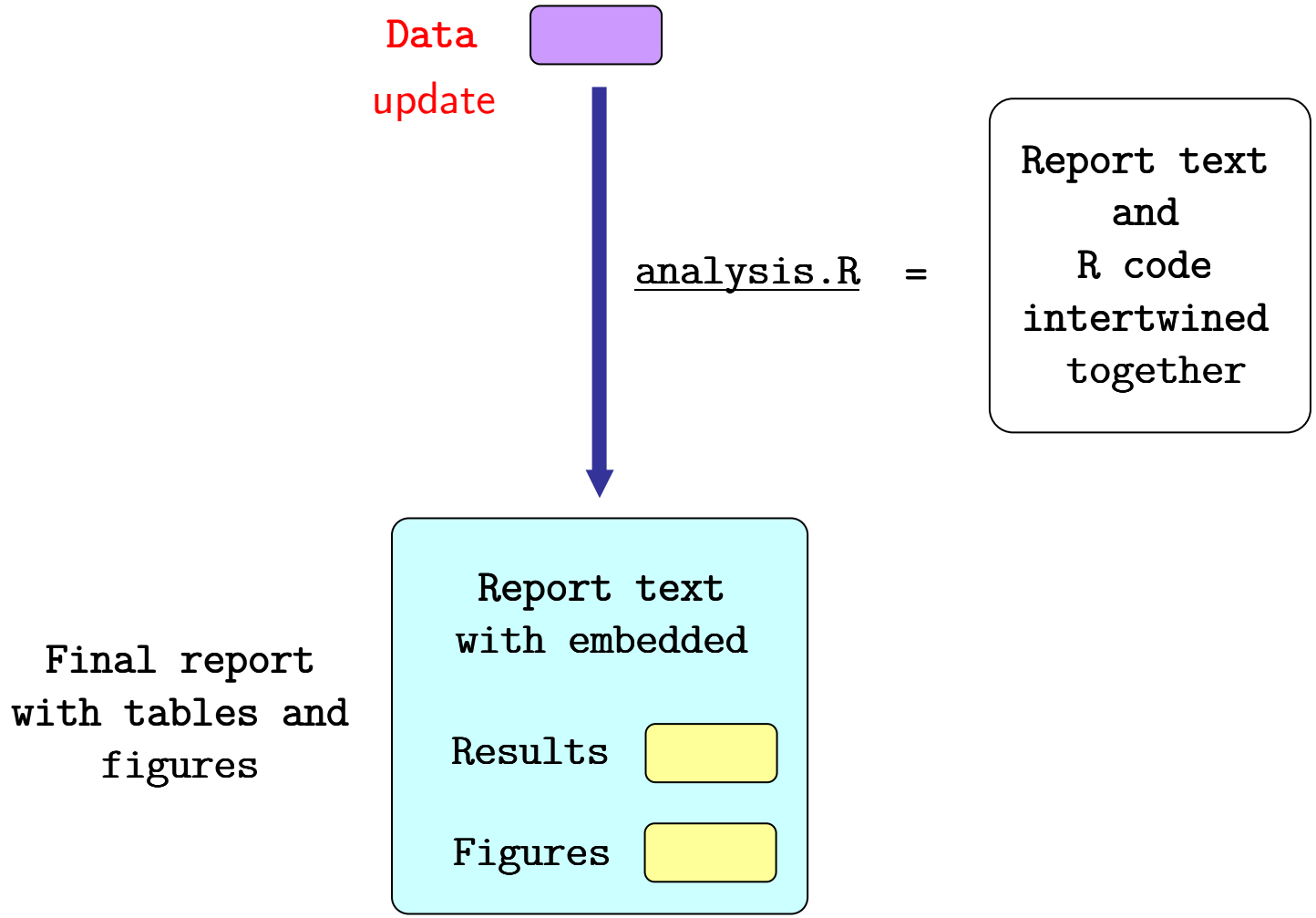
# Automatic report generation





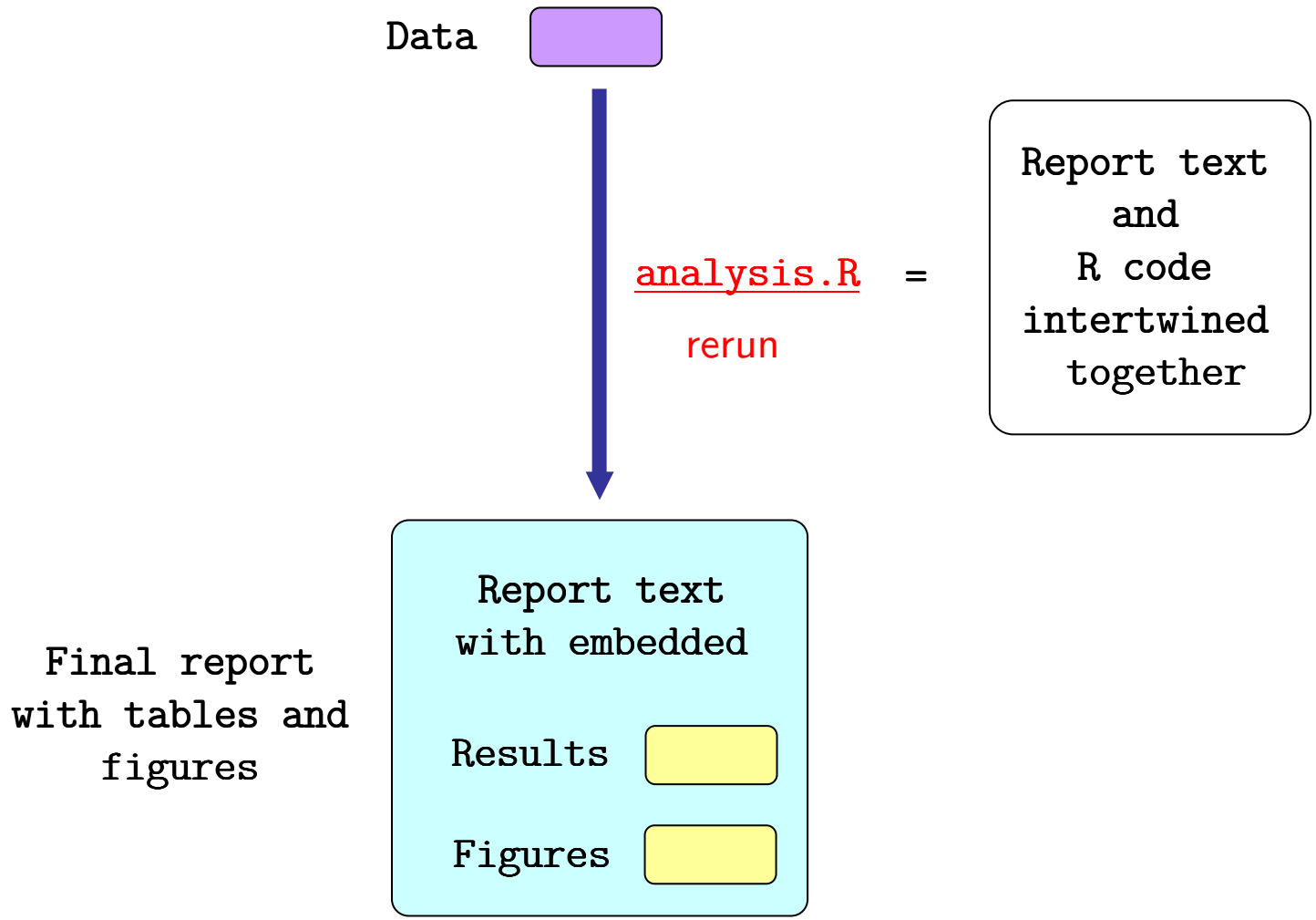
# Automatic report generation





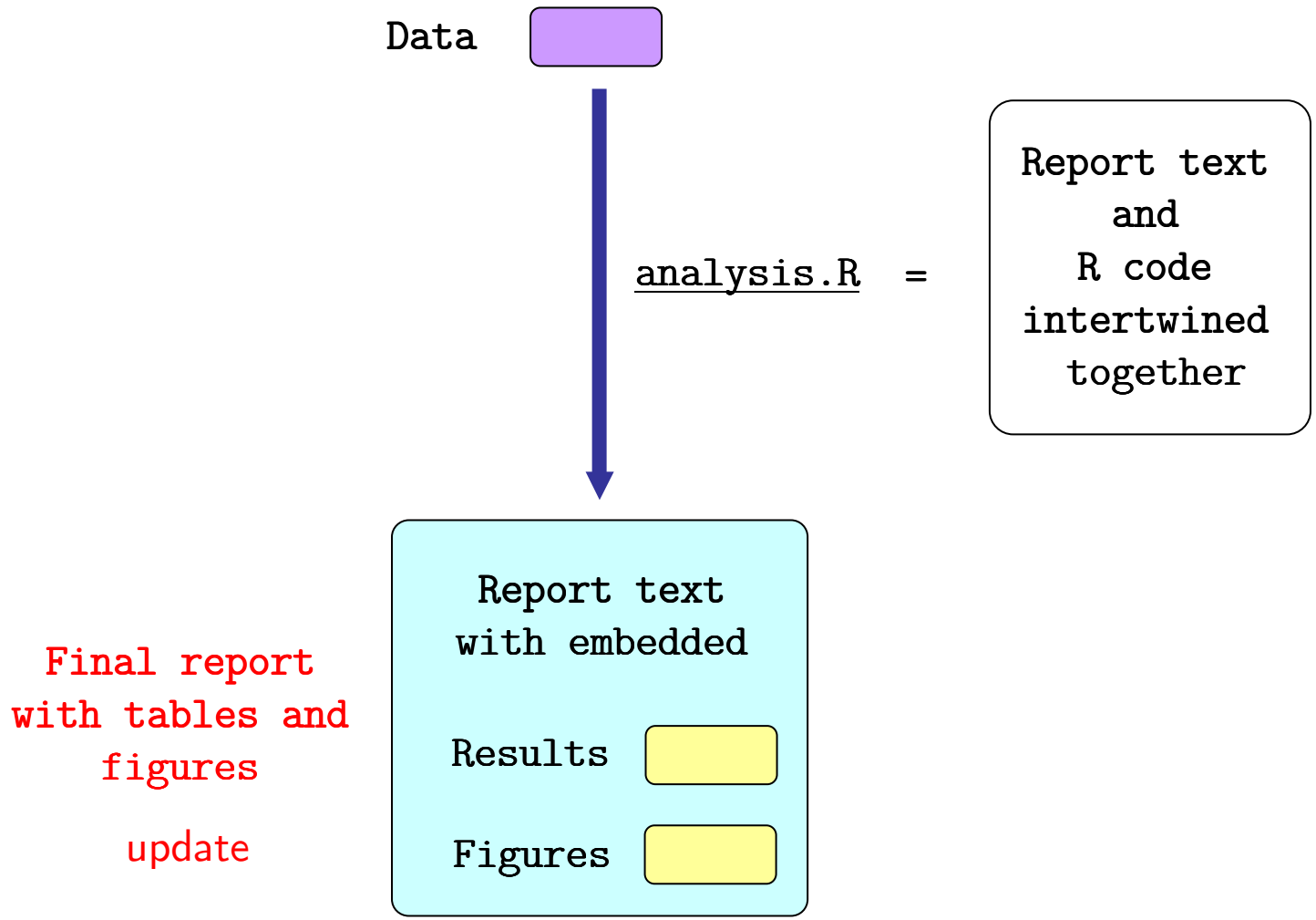
# Automatic report generation





# Automatic report generation





# Automatic report generation



- Data analysis, comments and interpretation are mixed together in the R script, as it is in reality.
- The final report is produced automatically and can be updated in a single step.
- The report generation script can be version controlled !

# Automatic report generation



- Data analysis, comments and interpretation are mixed together in the R script, as it is in reality.
- The final report is produced automatically and can be updated in a single step.
- The report generation script can be version controlled !

# Automatic report generation



- Data analysis, comments and interpretation are mixed together in the R script, as it is in reality.
- The final report is produced automatically and can be updated in a single step.
- The report generation script can be version controlled !

# Tools



- Scripts with R code and regular text mixed

````
Temperature evolution
=====================

We could study the temperature evolution of Turun Yliopisto
duck pond using publicly available oceanographic data.

This is the **first study** showing an increase in the duck pond
summer temperature over the last 20 years.

```{r fig.width=4, fig.height=4}
pond = read.table("pond_data", sep = "\t", header = T)
plot(pond$year, pond$temperature)
```
````

# Tools



- Scripts with R code and regular text mixed

Rich text

```
Temperature evolution
=====================

We could study the temperature evolution of Turun Yliopisto
duck pond using publicly available oceanographic data.

This is the **first study** showing an increase in the duck pond
summer temperature over the last 20 years.

```{r fig.width=4, fig.height=4}
pond = read.table("pond_data", sep = "\t", header = T)
plot(pond$year, pond$temperature)
```
```

# Tools



- Scripts with R code and regular text mixed

````
Temperature evolution
=====================

We could study the temperature evolution of Turun Yliopisto
duck pond using publicly available oceanographic data.

This is the **first study** showing an increase in the duck pond
summer temperature over the last 20 years.

```{r fig.width=4, fig.height=4}
pond = read.table("pond_data", sep = "\t", header = T)
plot(pond$year, pond$temperature)
```
````

R code

# Tools



- The script is interpreted with the regular text being formatted and the R code executed.

# Tools



- The script is interpreted with the regular text being formatted and the R code executed.

````
Temperature evolution
=====================

We could study the temperature evolution
of Turun Yliopisto duck pond using
publicly available oceanographic data.

This is the **first study** showing an
increase in the duck pond summer
temperature over the last 20 years.

```{r fig.witdh=4, fig.height=4}
pond = read.table("pond_data", sep = "\t",
header = T)
plot(pond$year, pond$temperature)
```
````

# Tools



- The script is interpreted with the regular text being formatted and the R code executed.

````
Temperature evolution
=====================

We could study the temperature evolution
of Turun Yliopisto duck pond using
publicly available oceanographic data.

This is the **first study** showing an
increase in the duck pond summer
temperature over the last 20 years.

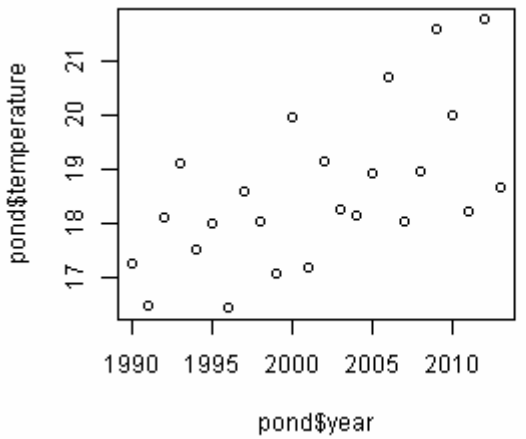
```{r fig.witdh=4, fig.height=4}
pond = read.table("pond_data", sep = "\t",
header = T)
plot(pond$year, pond$temperature)
```
````

## Temperature evolution

We could study the temperature evolution of Turun Yliopisto duck pond using publicly available oceanographic data.

This is the **first study** showing an increase in the duck pond summer temperature over the last 20 years.

```
pond = read.table("pond_data", sep = "\t", header = T)
plot(pond$year, pond$temperature)
```



# Tools




R script with
Markdown text
(.Rmd)
package
knitr
HTML output
RStudio
+
Git
R script with
LaTeX text
(.Rnw)
package
sweave
+
LaTeX
PDF output

# Tools



R script with
Markdown text
(.Rmd)

package
knitr

HTML output

RStudio
+
Git

R script with
LaTeX text
(.Rnw)



package
sweave
+
LaTeX

PDF output

# Tools



R script with Markdown text (.Rmd)

package knitr

HTML output

RStudio + Git

R script with LaTeX text (.Rnw)

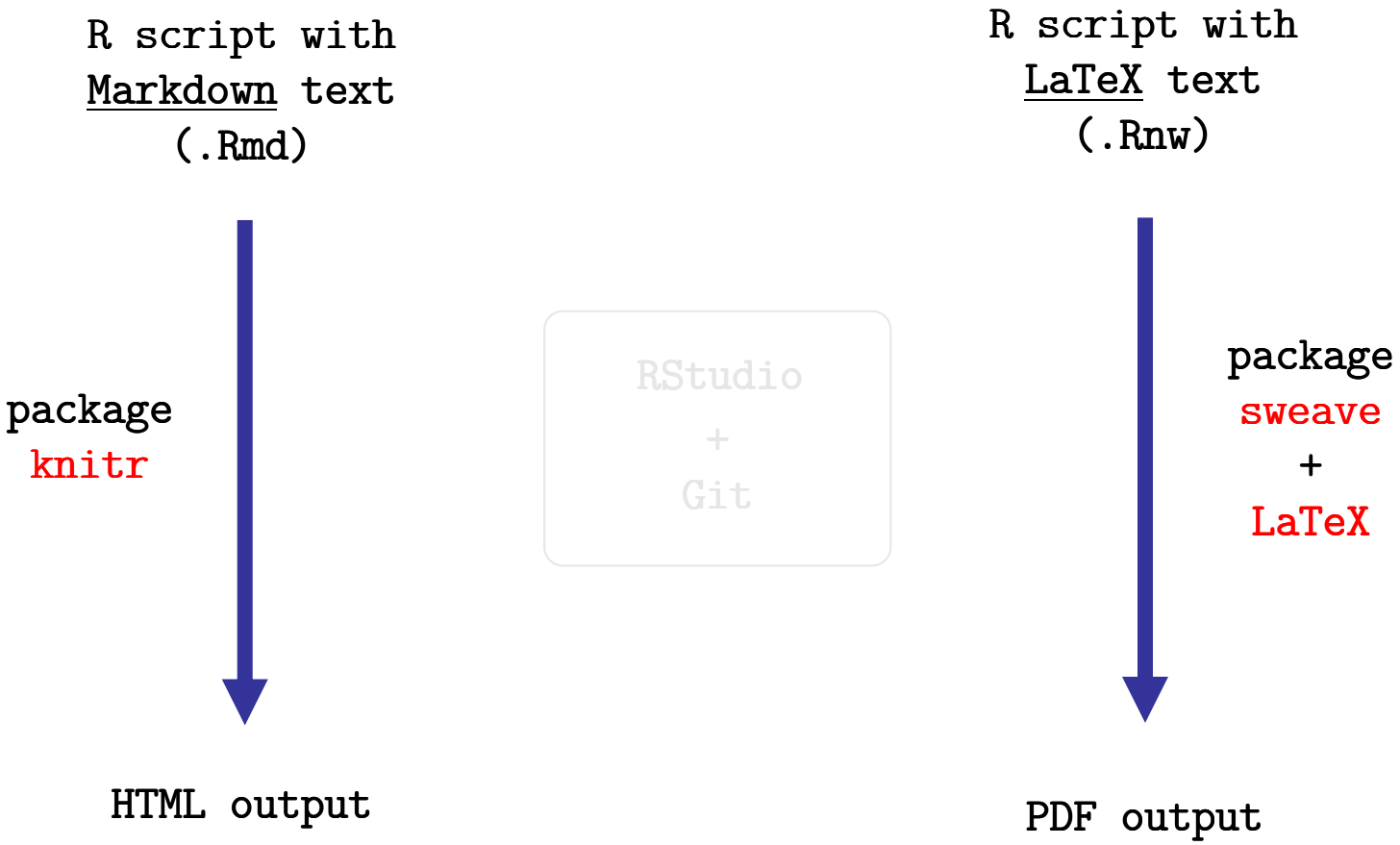


package sweave + LaTeX

PDF output

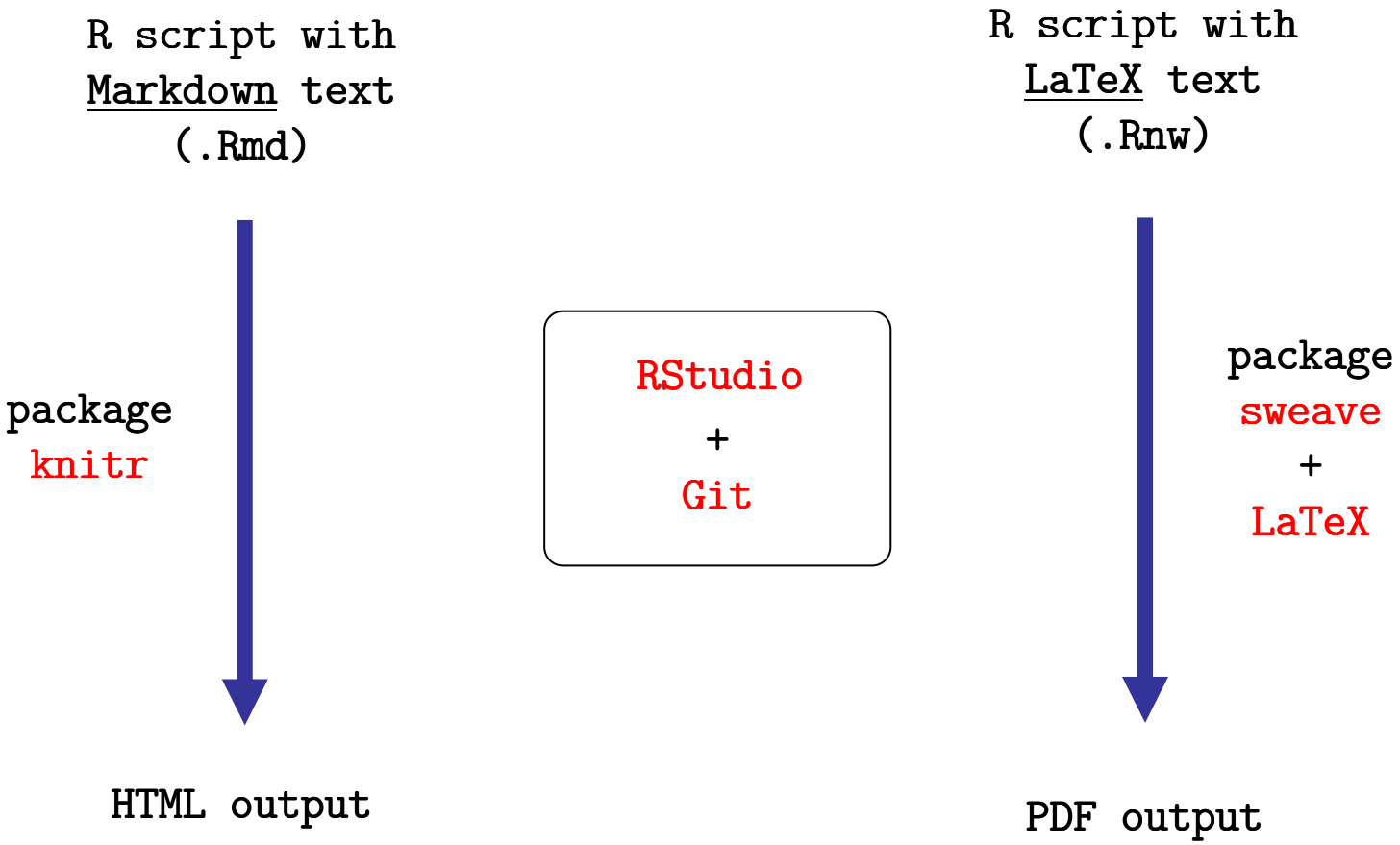
# Tools




R script with
Markdown text
(.Rmd)
package
knitr
HTML output
RStudio
+
Git
R script with
LaTeX text
(.Rnw)
package
sweave
+
LaTeX
PDF output

# Tools




R script with
Markdown text
(.Rmd)
package
knitr
HTML output
RStudio
+
Git
R script with
LaTeX text
(.Rnw)
package
sweave
+
LaTeX
PDF output

# Markdown - HTML output

# Markdown



- Markdown syntax is based on plain-text formatting commands (`*important*` = *important*)
- The resulting plain text remains easily human readable even without processing.
- R code chunks are embedded in the plain text and executed during processing.

# Markdown



- Markdown syntax is based on plain-text formatting commands (`*important*` = *important*)
- The resulting plain text remains easily human readable even without processing.
- R code chunks are embedded in the plain text and executed during processing.

# Markdown



- Markdown syntax is based on plain-text formatting commands (`*important*` = *important*)
- The resulting plain text remains easily human readable even without processing.
- R code chunks are embedded in the plain text and executed during processing.

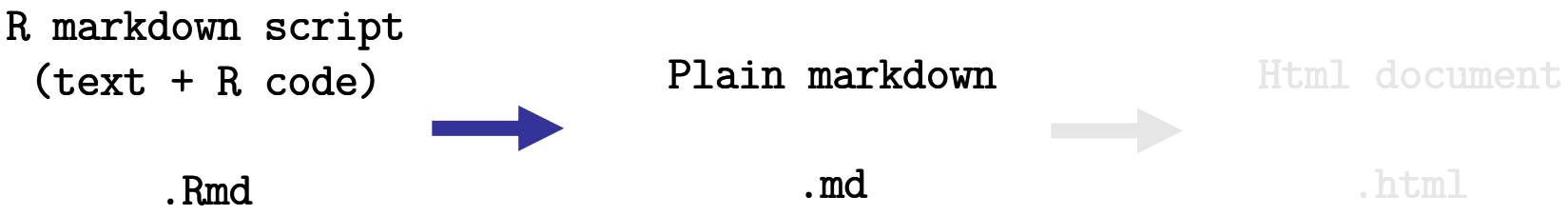
# Markdown




- RStudio with the `knitr` package take care of all the steps between the script and the html output.

# Markdown

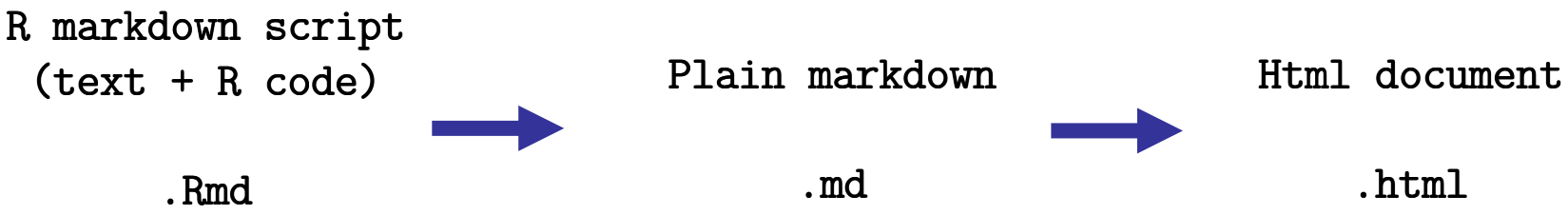





- RStudio with the `knitr` package take care of all the steps between the script and the html output.

# Markdown

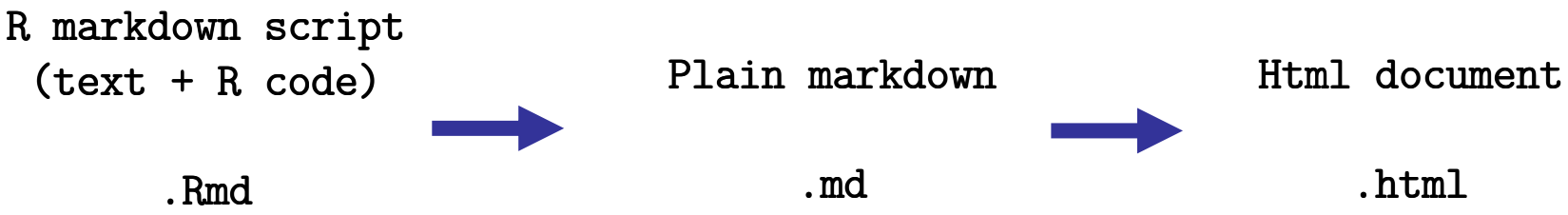





- RStudio with the `knitr` package take care of all the steps between the script and the html output.

# Markdown






- RStudio with the `knitr` package take care of all the steps between the script and the html output.

# Markdown - Practical

# Markdown



- We are going to analyse a Moomin related dataset to practice report generation with Markdown and Sweave + LaTeX.

- The exercise folder is `report_generation`. It contains a `data` folder with the original data files.

- In case of need, all the code used for the practical is available in the files:

  `moomin_reference.Rmd`
  `moomin_reference.Rnw`
  `moomin_reference_advanced.Rnw`

# Markdown



- First, let's start a new RStudio session.
- Create a new file by *'File > New > R Markdown'*. RStudio puts some code in it by default, erase it and save the empty file as `moomin.Rmd` in the `report_generation/markdown` folder (*'File > Save as...'*).
- Note the `MD` icon on top of the source pane. You can click on it for a markdown quick reference.

# Setting up the working directory



- In the *'Files'* tab, enter into the `report_generation/markdown` folder.
- Set it as the default directory by clicking *'More > Set as working directory'* on the top of the *'Files'* tab.
- This ensures that the relative paths to data files defined in the R code will be correct.

# Headers



- Headers can be specified easily by underlining some text.
- Write the following text in the source pane:

```
Effect of lakritsi on agressiveness: a Moomin perspective
=========================================================

Context of the study
--------------------

Haribo et al. have suggested a correlation between the
consumption of liquorice and agressive behaviour, but
their previous study on Smurfs remained inconclusive.
```

# Kniting



- You can knit your source file by clicking on *'Knit HTML'*.
- This will process the source file and open a preview of the html output.
- The icons at the top of the preview windows enable to open the document in the regular web browser, to see the compilation log or to save the document.

# Emphasis



- Text can be formatted to italics or bold with *.
- Modify your script this way and knit it:

```
Effect of lakritsi on agressiveness: a Moomin perspective
=========================================================

Context of the study
--------------------

Haribo *et al.* have suggested a correlation between the
consumption of **liquorice** and **agressive behaviour**, but
their previous study on Smurfs (*Schtroumpfus peyoii*) remained
inconclusive.
```

# Lists



- For lists, add this text to your script and knit it:

```
Haribo *et al.* study had 3 major flaws:
- they did not describe how they measured agressiveness
- most Smurfs escaped before the end of the experiment
- Smurfs have a low expression level of liquorice receptors, as
was later evidenced:
    1. in the brain
    2. in the liver
    3. in the muscle
```

# A bit more text



- Add more text to your script and knit it:

```
Experimental design
-------------------

We decided to test their hypothesis using a dataset collected on
 Moomins and other related characters. Our approach includes:
- male and female individuals from different species
- different levels of liquorice consumption among individuals
- paintball activity as a proxy for agressiveness
```

- Note how the plain-text of the source file is easily readable and already formatted by the markdown syntax itself.

# R code chunk



- Time for some code. Add this text to your script and knit it:

````
Individuals used in this study
------------------------------

```{r}
# Load Moomin data
moomin = read.table("../data/moomin_data", header = T, sep ="\t")
moomin
```
````

- Text delimited by ```` ```{r} ```` and ```` ``` ```` is interpreted as R code and executed when the file is processed. RStudio uses colored background to separate R code from plain-text.

# R code chunk



- Each R code block is called a code chunk. Code chunk output is added to the html document.
- Code chunks can be labelled for easy access. Modify your source this way:

````
Individuals used in this study
------------------------------

```{r loadMoominData}
# Load Moomin data
moomin = read.table("../data/moomin_data", header = T, sep ="\t")
moomin
```
````

# R code chunk



- Let's load a bit more data to continue our practice:

````
Results
-------

### Paintball activity

For each individual, the number of used balls during a 180min
paintball game was monitored.

```{r loadPaintballData}
# Load paintball data
paintball = read.table("../data/paintball_data", header = T,
                       sep ="\t")
head(paintball)
```
````

# Graphics



- We can easily add a plot to the output:

````
### Example of activity: Sniff (Nipsu)

This plot shows the amount of balls used by Sniff during the
game.

```{r plotSniffActivity}
# Extract Sniff activity
sniff_activity = subset(paintball, paintball$user == "Sniff")
plot(sniff_activity$time.min, sniff_activity$balls.used,
     type = "l")
```
````

# Graphics



- We can also use `ggplot`:

````
The next plot uses `ggplot2` for fancier graphics.

```{r plotSniffActivity.ggplot2}
library(ggplot2)
ggplot(sniff_activity, aes(x = time.min, y = balls.used)) +
   geom_line()
```
````

- Note the backquotes around ggplot to use the code font in the output.

# In-line code



- In-line code can be used and evaluated outside of R code chunks :

````
### Activity of all individuals

The next plot uses the data for all `r nrow(paintball)`
records in the paintball tables.

```{r plotActivityAll}
ggplot(paintball, aes(x = time.min, y = balls.used, col = user)) +
   geom_line()
```
````

- In-line code must be defined by typing `r` just after the opening backquote.

# R code chunk options



- R code chunk have many options that can be used. Let's make the previous plot a bit wider by modifying our code:

````
### Activity of all individuals

The next plot uses the data for all `r nrow(paintball_data)`
records in the paintball tables.

```{r plotActivityAll, fig.width = 8, fig.height = 5}
ggplot(paintball, aes(x = time.min, y = balls.used, col = user)) +
   geom_line()
```
````

# Hidden code



- We can run R code without displaying it in the final document. This can be useful for intermediate steps of limited interest for the result interpretation.

````
### Effect of lakritsi on paintball activity

We have an estimation of the average lakritsi consumption per
day for each individual. Consumption ranges from
`r min(moomin$liquorice.kg.day)` to
`r max(moomin$liquorice.kg.day)` kg per day.

```{r mergeTables, echo = FALSE}
all_data = merge(moomin, paintball, by.x = "english",
                 by.y = "user")
```
````

# Hidden code



- In the previously added text, note the use of in-line code for the liquorice consumption range. If the data is updated with new individuals, this range will be updated automatically when the report is generated again.
- Using echo=FALSE in the code chunk header makes the code invisible in the html output.
- At this point, note also how you can easily navigate between the code chunks using the menu at the bottom of the source pane.

# Hidden code



- Hidden code can be used to define functions or run lengthy code without crowding the output:

````
```{r calculateTotalNumberBalls, echo = FALSE}
total.n.balls = vector()
for (individual in moomin$english) {
  total = max(subset(paintball,
                     paintball$user == individual)$balls.used)
  total.n.balls = c(total.n.balls, total)
}
summary_moomin = data.frame(moomin$english,
                            moomin$liquorice.kg.day,
                            total.n.balls)
names(summary_moomin) = c("individual", "liquorice.kg.day",
                          "balls.game")
```
````

# More code



- We are basically interested in the effect of lakritsi on paintball activity. We can start by looking at it graphically:

````
The next plot uses color-coding to depict the average consumption
of lakritsi in kg/day.

```{r plotActivity.col.lakritsi, fig.width = 8, fig.height = 5}
ggplot(all_data, aes(x = time.min, y = balls.used, group = english)) +
   geom_line(aes(col = liquorice.kg.day))
```


We can also look at the total number of balls used in relation to the
liquorice consumption.

```{r plotTotalBallsLiquorice, fig.width = 8, fig.height = 5}
ggplot(summary_moomin, aes(x = liquorice.kg.day, y = balls.game,
                           col = individual)) + geom_point()
```
````

# Error handling



- Let's calculate some correlation coefficient:

````
The Spearman's rho test gives:

```{r SpearmanTest}
cor.test(summary_moomin$lakritsi, summary_moomin$balls.game,
         method = "spearman"
```
````

- A parenthesis is missing and knitr stops and gives us a hint about the location of the error.

# Error handling



- Let's correct and run it again:

````
The Spearman's rho test gives:

```{r SpearmanTest}
cor.test(summary_moomin$lakritsi, summary_moomin$balls.game,
         method = "spearman")
```
````

- Knitr runs fine but an error is visible on the html output. When clicking on the `log` icon, no more information is available since knitr ran correctly. The error was within the R code but was not a knitr related syntax error.

# Error handling



- We have to go back inside the corresponding chunk and correct it:

````
The Spearman's rho test gives:

```{r SpearmanTest}
cor.test(summary_moomin$liquorice.kg.day,
         summary_moomin$balls.game,
         method = "spearman")
```
````

- Now everything runs fine. A warning was produced by `cor.test` and it is included in the output.

# Tables



- Simple tables can also be inserted:

```
Comparison of the studies
-------------------------

Haribo *et al.* | our study
-----|-----
Smurfs | Moomins
unknown proxy | paintball activity
potential conflict of interest | no conflict of interest
```

# Running code chunks within RStudio



- Kniting the source file does not run it within RStudio: the variables are not defined and cannot be accessed from the console.
- Chunks have to be run manually, either one by one or all of them at the same time (menu *'Chunks'* on the top of the source pane).
- Typing `moomin` in the console gives an error. After running all the chuncks within RStudio, `moomin` returns the Moomin data frame. We can work as usual with RStudio.

# Summary



- R markdown is a very simple language with highly readable scripts.
- Its use ensures a tight link between the analysis and the report (the report file and the analysis file are the same).
- The scripts can be version controlled.

# Summary



- R markdown is a very simple language with highly readable scripts.
- Its use ensures a tight link between the analysis and the report (the report file and the analysis file are the same).
- The scripts can be version controlled.

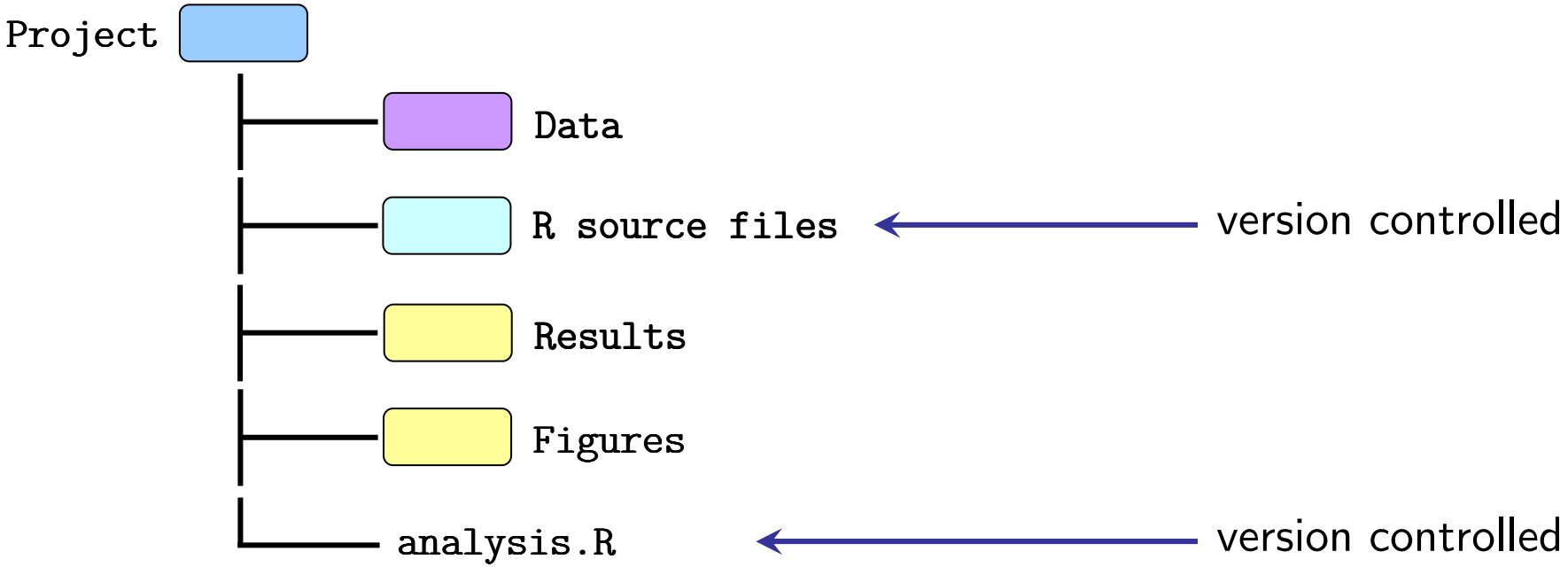
# Summary



- R markdown is a very simple language with highly readable scripts.
- Its use ensures a tight link between the analysis and the report (the report file and the analysis file are the same).
- The scripts can be version controlled.

# Examples of project folder

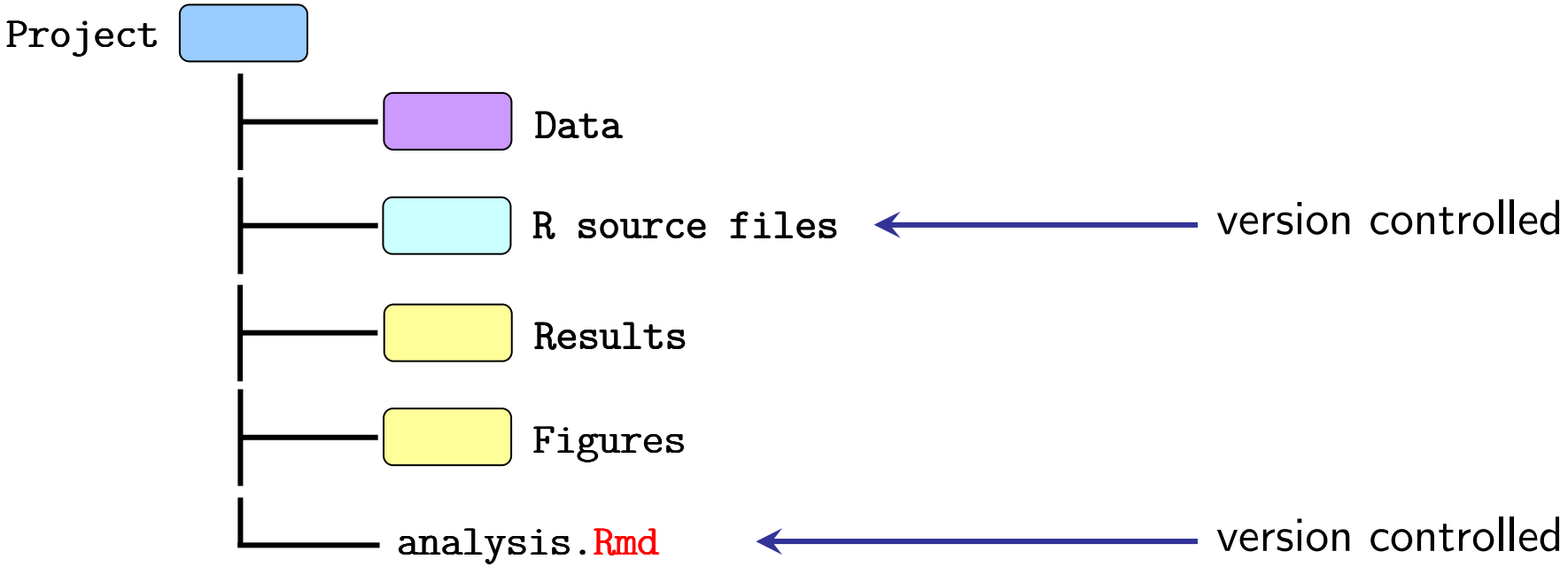





(shamelessly modified from http://nicercode.github.io/blog/2013-04-05-projects/)

# Examples of project folder

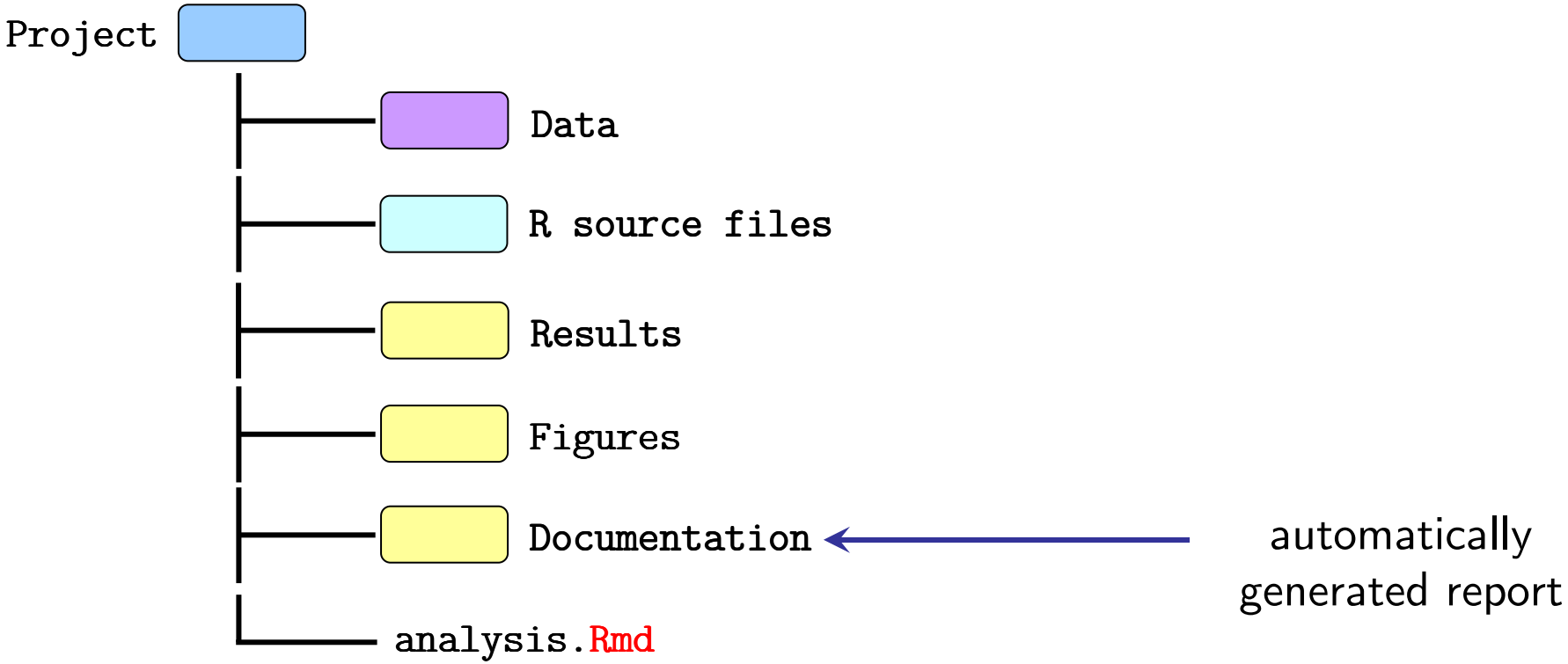





(shamelessly modified from http://nicercode.github.io/blog/2013-04-05-projects/)

# Examples of project folder






(shamelessly modified from http://nicercode.github.io/blog/2013-04-05-projects/)

# Sweave + LaTex - PDF output

# Sweave + LaTex



- LaTeX is a software designed for typesetting high-quality documents.
- It is very powerful and versatile but the corresponding learning curve is steep.
- It is widely used in mathematics and physics.

# Sweave + LaTex



- LaTeX is a software designed for typesetting high-quality documents.
- It is very powerful and versatile but the corresponding learning curve is steep.
- It is widely used in mathematics and physics.

# Sweave + LaTex



- LaTeX is a software designed for typesetting high-quality documents.
- It is very powerful and versatile but the corresponding learning curve is steep.
- It is widely used in mathematics and physics.

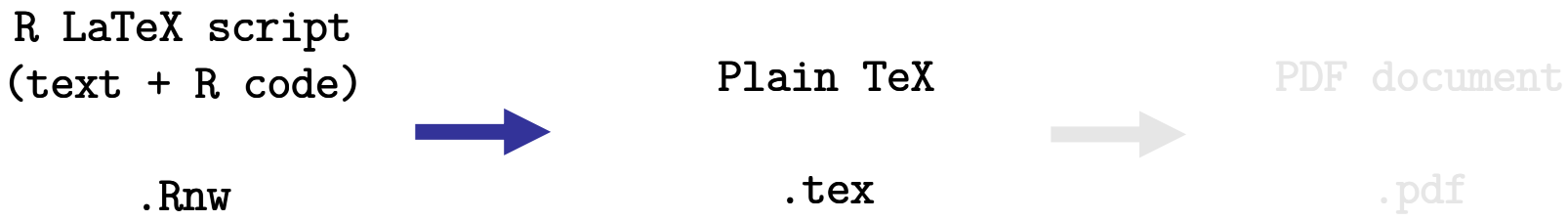
# Sweave + LaTeX




- RStudio with the Sweave package and a LaTex distribution take care of all the steps between the script and the pdf output.

# Sweave + LaTeX

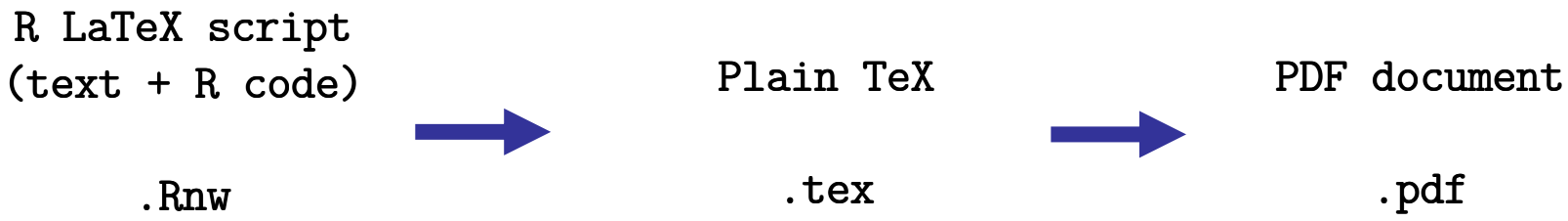





- RStudio with the Sweave package and a LaTex distribution take care of all the steps between the script and the pdf output.

# Sweave + LaTeX

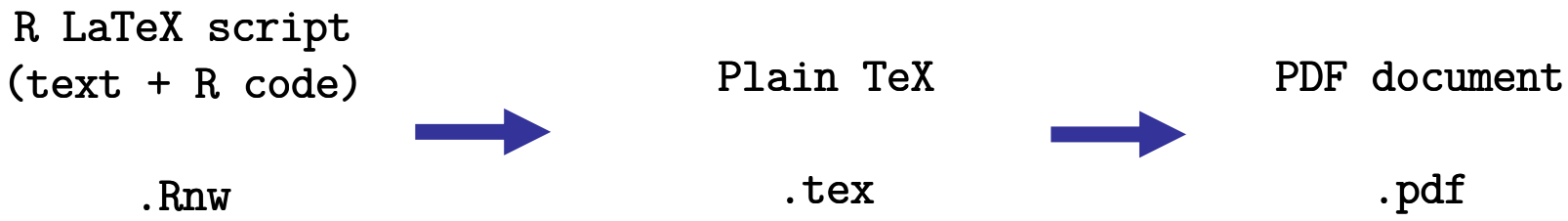





- RStudio with the Sweave package and a LaTex distribution take care of all the steps between the script and the pdf output.

# Sweave + LaTeX






- RStudio with the `Sweave` package and a LaTeX distribution take care of all the steps between the script and the pdf output.

# Disclaimer



- Sweave is very versatile but is not always very comfortable to use.
- knitr has simplified many aspects of using LaTeX for R report generation.
- The practical focuses on Sweave since I know it, but if you are new to it you might consider learning how to use knitr rather than Sweave.

# Disclaimer



- Sweave is very versatile but is not always very comfortable to use.
- knitr has simplified many aspects of using LaTeX for R report generation.
- The practical focuses on Sweave since I know it, but if you are new to it you might consider learning how to use knitr rather than Sweave.

# Disclaimer



- Sweave is very versatile but is not always very comfortable to use.
- knitr has simplified many aspects of using LaTeX for R report generation.
- The practical focuses on Sweave since it is what I know, but if you are new to it you might consider learning how to use knitr rather than Sweave.

# Sweave + LaTeX - Practical

# Sweave + LaTeX



- We are going to analyse the same Moomin related dataset as before to practice report generation with Sweave + LaTeX.
- LaTeX itself must be installed on the computer. A LaTeX distribution is usually quite large (>150 Mb for Windows).
- The LaTeX distribution installation folder should be in the PATH environment variable for Windows OS.

# Modifying the PATH (Windows 7)



- Press the Windows key and the Pause key simultaneously.
- Click *'Advanced parameter settings > Environment variables > Path > Edit...'*.
- Add the path corresponding to your LaTeX distribution, separated from the previous path by a semicolon.
- e.g.: C:\Program Files\MiKTeX 2.9\miktex\bin
- Click *'Ok'*.

# Sweave + LaTeX



- Let's start by closing all the source files in RStudio and create a new file by clicking *'File > New > R Sweave'*.
- Erase the R default code and save the empty file as `Moomin.Rnw` in the `report_generation/sweave` folder.
- Remember that you can use the file `moomin_reference.Rnw` if you need it at some point during the practical.

# Setting up the working directory



- In the *'Files'* tab, enter into the `report_generation/sweave` folder.
- Set it as the default directory by clicking *'More > Set as working directory'* on the top of the *'Files'* tab.
- This ensures that the relative paths to data files defined in the R code will be correct.

# Document structure



- Every LaTeX document has a class (e.g. article, book).
- All the printed content must be contained in the `document` environment, i.e. between the two commands `\begin{document}` and `\end{document}`.
- Let's type our very first LaTeX code and save it:

```
\documentclass{article}

\begin{document}

This is a draft about Moomin.

\end{document}
```

# Document structure



- Click on *'Compile PDF'* to start the compilation. The first time, several additional packages might have to be installed by the LaTeX distribution.

- A pdf file with only one page is created.

# Title page



- We can create a title page with a few simple commands:

```
\documentclass{article}

\title{Effect of lakritsi on agressiveness: a Moomin perspective}
\author{Matthieu Bruneaux}

\begin{document}

\maketitle

This is a draft about Moomin.

\end{document}
```

# Title page



- We specify some parameters with `\title` and `\author`, while the `\maketitle` command inside the `document` environment automatically generates a title page.
- Let's save it and compile it.
- The date is automatically added when the title page is built.

# Sections



- A LaTeX document is divided into chapters, sections, subsections and paragraphs.
- For the `article` class, the chapter subdivision level is not available.
- Let's add the first section:

```
\begin{document}

\maketitle

\section{Context of the study}

This is a draft about Moomin.

\end{document}
```

# Text



- We replace the draft text by the real one:

```
\begin{document}

\maketitle

\section{Context of the study}

Haribo et al. have suggested a correlation between the
consumption of liquorice and agressive behaviour, but
their previous study on Smurfs (Schtroumpfus peyoii) remained
inconclusive.

\end{document}
```

# Text formatting



- Bold font can be obtained with `\textbf{}`:

```
\begin{document}

\maketitle

\section{Context of the study}

Haribo et al. have suggested a correlation between the
consumption of \textbf{liquorice} and \textbf{agressive behaviour},
but their previous study on Smurfs (Schtroumpfus peyoii) remained
inconclusive.

\end{document}
```

# Text formatting



- Italics can be obtained with `\textit{}`:

```
\begin{document}

\maketitle

\section{Context of the study}

Haribo et al. have suggested a correlation between the
consumption of \textbf{liquorice} and \textbf{agressive behaviour},
but their previous study on Smurfs (\textit{Schtroumpfus peyoii})
remained inconclusive.

\end{document}
```

# Lists



- Lists are more complicated and need a special `itemize` environment. Each element of the list is defined by the `\item` command.

```
Haribo et al. study had 3 major flaws:

\begin{itemize}

\item they did not describe how they measured agressiveness
\item most Smurfs escaped before the end of the experiment
\item Smurfs have a low expression level of liquorice receptors, as
was later evidenced:

\end{itemize}
```

# Lists



- Numbered lists can be obtained with the `enumerate` environment.

```
Haribo et al. study had 3 major flaws:

\begin{itemize}

\item they did not describe how they measured agressiveness
\item most Smurfs escaped before the end of the experiment
\item Smurfs have a low expression level of liquorice receptors, as
was later evidenced:

\begin{enumerate}
\item in the brain
\item in the liver
\item in the muscle
\end{enumerate}

\end{itemize}
```

# Lists



- Indentation is not necessary but is helpful for the eye:

```
Haribo et al. study had 3 major flaws:

\begin{itemize}

    \item they did not describe how they measured agressiveness
    \item most Smurfs escaped before the end of the experiment
    \item Smurfs have a low expression level of liquorice
    receptors, as
    was later evidenced:

    \begin{enumerate}
        \item in the brain
        \item in the liver
        \item in the muscle
    \end{enumerate}

\end{itemize}
```

# More text



- Let's add the following section:

```
\section{Experimental design}

We decided to test their hypothesis using a dataset collected on
Moomins and other related characters. Our approach includes:

\begin{itemize}
    \item male and female individuals from different species
    \item different levels of liquorice consumption among
    individuals
    \item paintball activity as a proxy for agressiveness
\end{itemize}
```

# R code chunk



- With Sweave, R code is also contained in chunks delimited by <<>>= and @:

```
\section{Individuals used in this study}

<<>>=
# Load Moomin data
moomin = read.table("../data/moomin_data", header = T, sep ="\t")
moomin
@
```

- Labels can be added to the chunks:

```
<<label=loadMoominData>>=
# Load Moomin data
moomin = read.table("../data/moomin_data", header = T, sep ="\t")
moomin
@
```

# Comments



- LaTeX comments can be added by using % at the beginning of a line, outside code chunks.

- Indentation, empty lines and plain-text underlining using comment lines can help to make the source file more human readable.

```
\section{Experimental design}
%---------------------------

We decided to test their hypothesis using a dataset collected on
Moomins and other related characters. Our approach includes:

\begin{itemize}

  \item male and female individuals from different species
  \item different levels of liquorice consumption among individuals
  \item paintball activity as a proxy for agressiveness

\end{itemize}
```

# More text and code



- Let's add the next section and subsection:

```
\section{Results}

\subsection{Paintball activity}

For each individual, the number of balls used during a 180min
paintball game was monitored.

<<label=loadPaintballData>>=
# Load paintball data
paintball = read.table("../data/paintball_data", header = T,
                       sep ="\t")
head(paintball)
@
```

# Graph



- One figure can be produced per code chunk. `fig` and `include` options have to be set to `TRUE` for this chunk.

```
\subsection{Example of activity: Sniff (Nipsu)}

This plot shows the amount of balls used by Sniff during the
game.

<<label=plotSniffActivity, fig=TRUE, include=TRUE>>=
# Extract Sniff activity
sniff_activity = subset(paintball, paintball$user == "Sniff")
plot(sniff_activity$time.min, sniff_activity$balls.used, type =
"l")
@
```

# Graph



- We can also use ggplot, but we need to force the plotting by using a `print` command.

```
The next plot uses `ggplot2` for fancier graphics.

<<label=plotSniffActivity-ggplot2, fig=TRUE, include=TRUE>>=
library(ggplot2)
graph = ggplot(sniff_activity, aes(x = time.min, y = balls.used)) +
     geom_line()
print(graph)
@
```

# In-line code



- In-line code can be evaluated using `\Sexpr{}`.

```
\subsection{Activity of all individuals}

The next plot uses the data for all \Sexpr{nrow(paintball)}
records in the paintball tables.

<<label=plotActivityAll, fig=TRUE, include=TRUE, width = 8, height
= 6>>=
graph = ggplot(paintball, aes(x = time.min, y = balls.used,
                              col = user)) +
   geom_line()
print(graph)
@
```

# More in-line code



- A bit more of in-line code which would be updated if the data was to change:

```
\subsection{Effect of lakritsi on paintball activity}

We have an estimation of the average lakritsi consumption per
day for each individual. Consumption ranges from
\Sexpr{min(moomin$liquorice.kg.day)} to
\Sexpr{max(moomin$liquorice.kg.day)} kg per day.
```

# Hidden code



- Code can be omitted from the output by using `echo=FALSE`. Useful for lengthy or "boring" code.

```
<<label=mergeTables, echo=FALSE>>=
all_data = merge(moomin, paintball, by.x = "english",
                 by.y = "user")
@

<<label=calculateTotalNumberBalls, echo=FALSE>>=
total.n.balls = vector()
for (individual in moomin$english) {
  total = max(subset(paintball,
                     paintball$user == individual)$balls.used)
  total.n.balls = c(total.n.balls, total)
}
summary_moomin = data.frame(moomin$english,
                            moomin$liquorice.kg.day,
                            total.n.balls)
names(summary_moomin) = c("individual", "liquorice.kg.day",
                          "balls.game")
@
```

# More code



- Again, let's look at the effect of lakritsi on paintball activity.

```
The next plot uses color-coding to depict the average consumption
of lakritsi in kg/day.

<<label=plotActivity-col-lakritsi, fig=TRUE, width = 8, height =
5>>=
print(ggplot(all_data, aes(x = time.min, y = balls.used,
                           group = english)) +
      geom_line(aes(col = liquorice.kg.day)))
@
```

# More code



- And let's see the global trend:

```
We can also look at the total number of balls used in relation to
the liquorice consumption.

<<label=plotTotalBallsLiquorice, fig=TRUE, width = 8, height = 5>>=
print(ggplot(summary_moomin, aes(x = liquorice.kg.day,
                                 y = balls.game,
                                 col = individual)) +
      geom_point())
@
```

# Error handling



- Let's calculate some correlation coefficient:

```
The Spearman's rho test gives:

<<label=SpearmanTest>>=
cor.test(summary_moomin$lakritsi, summary_moomin$balls.game,
         method = "spearman"
@
```

- A parenthesis is missing and the compilation stops. RStudio gives us a hint about the location of the error (we can click on *'output'* to have more details).

# Error handling



- Let's fix the parenthesis:

```
The Spearman's rho test gives:

<<label=SpearmanTest>>=
cor.test(summary_moomin$lakritsi, summary_moomin$balls.game,
         method = "spearman"
@
```

- Again, the compilation stops. We can have information about the error as previously for the missing parenthesis.

# Error handling



- Let's fix the column name:

```
The Spearman's rho test gives:

<<label=SpearmanTest>>=
cor.test(summary_moomin$liquorice.kg.day,
         summary_moomin$balls.game,
         method = "spearman")
@
```

- Again, the compilation stops. `cor.test` runs fine but inserts a tab character in the output that bothers LaTeX.

# Error handling



- It means we have to manually get the results:

```
The Spearman's rho test gives:

<<label=SpearmanTest>>=
a = cor.test(summary_moomin$liquorice.kg.day,
         summary_moomin$balls.game,
         method = "spearman")
a$p.value
a$estimate
@
```

# Tables



- Tables can easily be printed using the `xtable` package (installed by typing `install.packages("xtable")`):

```
\section{Comparison of the studies}

<<label=comparison>>=
library(xtable)
theirs = c("Smurfs", "unknown proxy", "potential conflict of
interest")
ours = c("Moomins", "paintball activity", "no conflict of
interest")
comparison = data.frame(Haribo.et.al = theirs, our.study = ours)
print(xtable(comparison, caption = "Comparison of the studies"))
@
```

# Scratching the surface of LaTeX...



- Reference code for the rest of this practical is available in the file `moomin_reference_advanced.Rnw`

- A table of contents can be automatically inserted with the `\tableofcontents` command:

```
\maketitle

\tableofcontents

\section{Context of the study}
```

# Scratching the surface of LaTeX...



- Idem for the lists of figures and tables with the `\listoffigures` and `\listoftables` commands:

```
\maketitle

\tableofcontents

\listoffigures

\listoftables

\section{Context of the study}
```

- The list of figures is empty, we will see why and fix it later.

# Scratching the surface of LaTeX...



- We can add hyperlinks within the document with the `hyperref` LaTeX package:

```
\documentclass{article}

\usepackage[pdfborder={0 0 0}]{hyperref}

\title{Effect of lakritsi on agressiveness: a Moomin perspective}
```

- We can the document margins with the `geometry` LaTeX package:

```
\usepackage[margin=3.5cm]{geometry}
```

# Scratching the surface of LaTeX...



- For LaTeX to be aware of the figures, we must insert them in a `figure` environment:

```
\begin{figure}
\begin{center}
<<label=plotSniffActivity, fig=TRUE, echo=FALSE>>=
# Extract Sniff activity
sniff_activity = subset(paintball, paintball$user == "Sniff")
plot(sniff_activity$time.min, sniff_activity$balls.used, type =
"l")
@
\end{center}
\caption{Example of individual activity: Sniff}
\label{fig.Sniff.activity}
\end{figure}
```

# Scratching the surface of LaTeX...



- The label from the `figure` environment enables to refer to it from another part of the text:

```
The plot in Figure \ref{fig.Sniff.activity} shows the amount of
balls used by Sniff during the game.
```

- LaTeX takes care of placing the figures and updating the references and links. However, this means the figure can be placed before or after the text referring to it.

# Scratching the surface of LaTeX...



- LaTeX can also handle bibliography:

```
Haribo et al. \cite{haribo_liquorice_1973} have suggested a
correlation between the consumption of \textbf{liquorice} and
\textbf{agressive behaviour}, but their previous study on Smurfs
(\textit{Schtroumpfus peyoii}) remained inconclusive.
```

- At the end of the file:

```
\bibliographystyle{unsrt}

\bibliography{myLibrary}

\end{document}
```

- LaTeX will load the bibliography from `myLibrary.bib`.

# Summary



- LaTeX is powerful but harder to learn than markdown.
- The plain-text files tend to be less human readable.
- LaTeX provides tools for more elaborated documents with table of contents and lists of figures, hyperlinks, automatic numbering and bibliography.

# Summary



- LaTeX is powerful but harder to learn than markdown.
- The plain-text files tend to be less human readable.
- LaTeX provides tools for more elaborated documents with table of contents and lists of figures, hyperlinks, automatic numbering and bibliography.

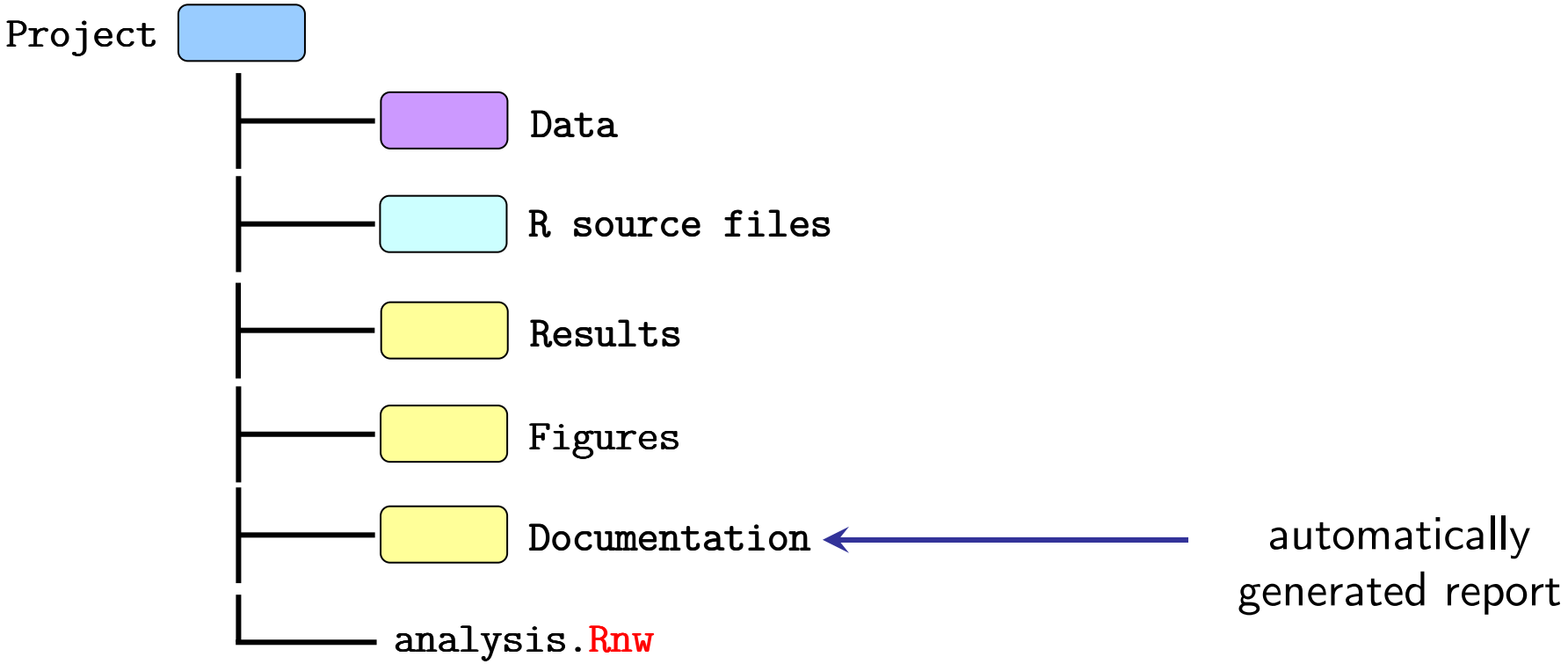
# Summary



- LaTeX is powerful but harder to learn than markdown.
- The plain-text files tend to be less human readable.
- LaTeX provides tools for more elaborated documents with table of contents and lists of figures, hyperlinks, automatic numbering and bibliography.

# Examples of project folder

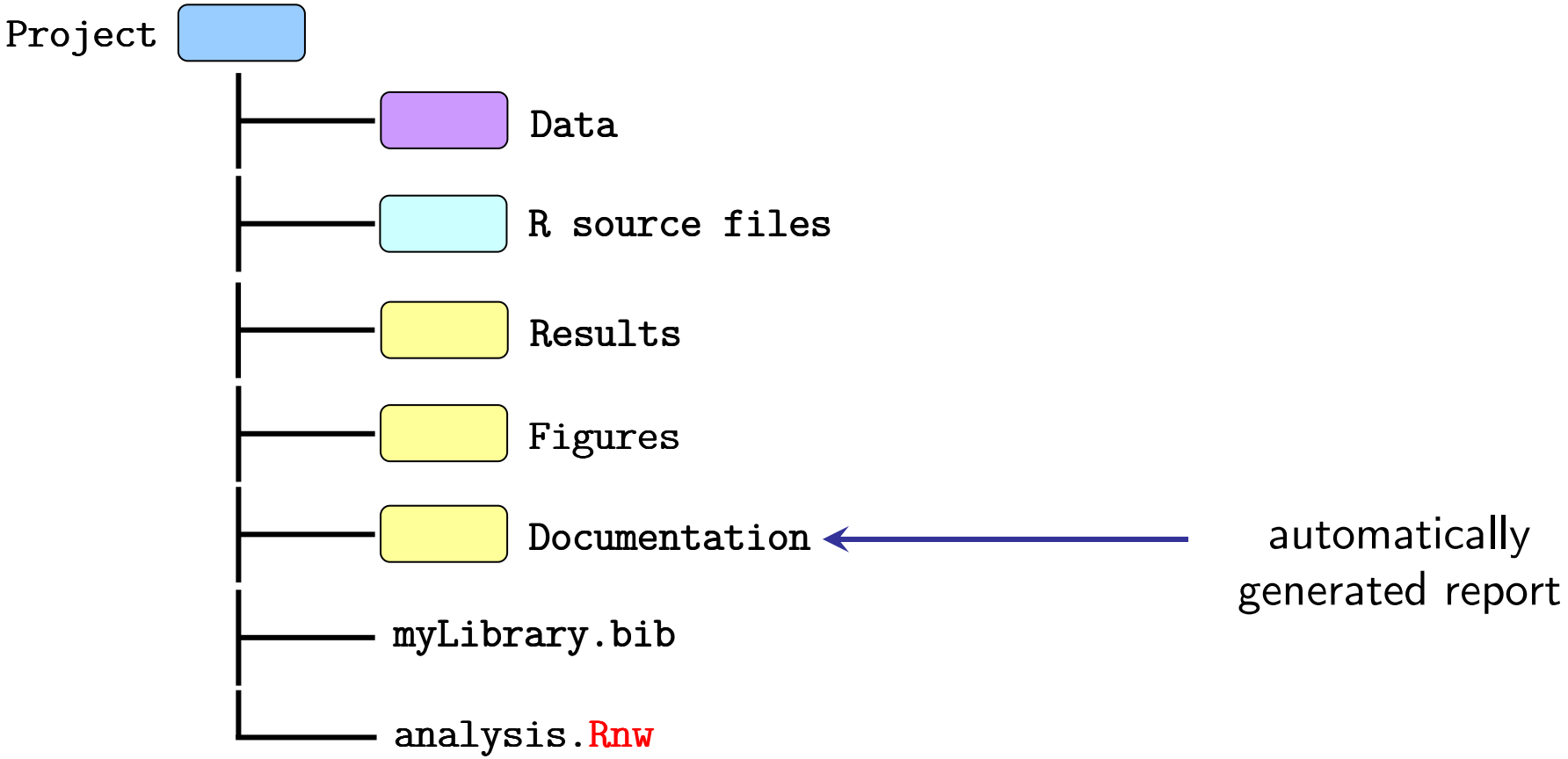





(shamelessly modified from http://nicercode.github.io/blog/2013-04-05-projects/)

# Examples of project folder

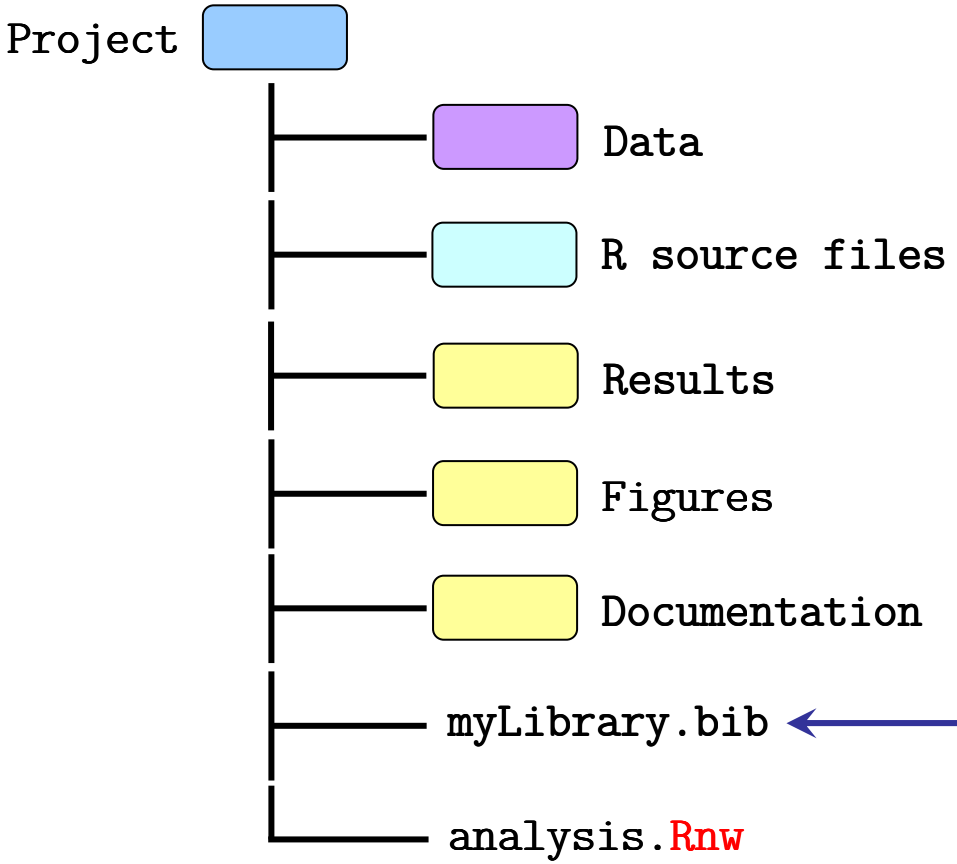





(shamelessly modified from http://nicercode.github.io/blog/2013-04-05-projects/)

# Examples of project folder






if needed;
can be generated
by bibliography
managers

(shamelessly modified from http://nicercode.github.io/blog/2013-04-05-projects/)

# In brief



- Report generation guarantees that the results you are looking at have been generated by the exact code embedded in the same file.
- A report is updated by running the source file when data or analysis code change.
- Associating version control and report generation ensures having up-to-date documents, where everything is (in theory) traceable.

# In brief



- Report generation guarantees that the results you are looking at have been generated by the exact code embedded in the same file.
- A report is updated by running the source file when data or analysis code change.
- Associating version control and report generation ensures having up-to-date documents, where everything is (in theory) traceable.

# In brief



- Report generation guarantees that the results you are looking at have been generated by the exact code embedded in the same file.
- A report is updated by running the source file when data or analysis code change.
- Associating version control and report generation ensures having up-to-date documents, where everything is (in theory) traceable.

# Further reading and references

# Further reading and references

## RStudio documentation

- Links for Sweave, knitr and R markdown documentation:
  http://rstudio.com/ide/docs/

## knitr

- Project homepage with lots of resources:
  http://yihui.name/knitr/

## Tutorials can easily be found on the internet.

# Next Part
# Figure generation with `make`